Representational similarity modulates neural and behavioral signatures of novelty

Sophia Becker[1,2], Alireza Modirshanechi[3,4], and Wulfram Gerstner[1,2]

[1]School of Computer and Communication Sciences, EPFL
[2]School of Life Sciences, EPFL
[3]Helmholtz Munich
[4]MPI for Biological Cybernetics

December 10, 2024

Abstract

Novelty signals in the brain modulate learning and drive exploratory behaviors in humans and animals. While the perceived novelty of a stimulus is known to depend on previous experience, the effect of stimulus representations on novelty computation remains elusive. In particular, existing models of novelty computation fail to account for the effects of stimulus similarities that are abundant in naturalistic environments and tasks. Here, we present a unifying, biologically plausible model that captures how stimulus representations modulate novelty signals in the brain and influence novelty-driven learning and exploration. By applying our model to two publicly available data sets, we quantify and explain (i) how generalization across similar visual stimuli affects novelty responses in the mouse visual cortex, and (ii) how generalization across nearby locations impacts mouse exploration in an unfamiliar environment. Our model unifies and explains distinct neural and behavioral signatures of novelty, and enables theory-driven experiment design to investigate the neural mechanisms of novelty computation.

Introduction

Novelty signaling in the brain is vital to facilitate learning [1–7], enhance sensory processing [8–12] and drive behavior [13–21]. Many experimental studies use computational models of novelty to explain and predict the effects of novelty signaling on neural activity and behavior [16–20, 22–24].

However, the concept of novelty and its quantification in these studies relies on the assumption that stimuli are either 'identical' or 'different'. This approach ignores any further similarities between stimuli and causes substantial limitations in environments that are continuous or exhibit stimulus similarities. For example, consider two previously unobserved stimuli that are similar to each other but not identical, e.g. two paintings by the same artist that exhibit a similar painting technique and style. If the two stimuli are presented sequentially, then how 'novel' is the second painting after observing the first one? Classic novelty models force us to either (i) consider the two stimuli as separate entities, implying that the second painting is completely novel despite its similarity to the first painting that was just observed; or (ii) consider them as the 'same' stimulus, predicting that no matter whether we continue observing the first or the second painting, we get equally familiar with both of them. Both options are at odds with what we would expect from human and animal perception (see e.g. [25] for a review). In the same way, classic novelty models fail to address novelty computation in continuous environments, e.g. during spatial exploration. Indeed, several experiments suggest that the brain's novelty response to a given stimulus does not merely depend on the number of times the exact same stimulus has been observed, but is also modulated by exposure to specific features of parts of the stimulus [13, 26–28] (see Discussion). This underlines the conceptual limitations of classic novelty models and raises the question how computational models of novelty can account for stimulus similarities.

In the field of machine learning, novelty-like signals are used to guide artificial agents during exploration of unfamiliar environments with sparse or no rewards [29–35]. These novelty-like signals are often computed as decreasing functions of the observed frequency of a given input ('state') to the artificial agent. The observed frequency is computed by neural networks that, through training, generalize across similar states [29–35]. The success of such novelty-guided artificial agents in exploring complex environments highlights the functional relevance of generalization across similar stimuli or similar locations in the environment. However, while machine-learning models of novelty computation deal well with continuous and complex environments, they rely on extensive training of separate networks to estimate novelty, as well as architectures and learning rules with limited biological plausibility and interpretability.

Here, we propose a unified modeling framework for novelty computation ('similarity-based novelty') that leverages probabilistic mixture models to combine the strengths of classic and ML-inspired novelty models. Our model is consistent with classic novelty models [16–20,22–24] (see [36] for a review) in environments with discrete and distinct stimuli, but it also extends to continuous spaces and environments with similarity structure. Our model implements novelty updates as a biologically interpretable plasticity rule that is consistent with various circuit models of novelty processing (e.g., based on adaptation, Hebbian synaptic plasticity or inhibitory circuits [37–40]) and allows to test experimental hypotheses about how novelty-computing circuits represent stimuli and stimulus similarities. We test our similarity-based novelty model in two data sets and derive new experimental predictions.

Results

Count-based novelty models fail to account for similarity modulation of novelty responses

Most algorithmic models of novelty computation in the brain [16–20, 22–24, 41, 42] are based on estimating the count-based frequency at which stimuli (e.g., sensory stimuli, spatial locations etc.) have been observed: The more often a stimulus is observed, the less novel it is. These 'count-based' novelty models usually consist of four building blocks: (i) a stimulus representation, (ii) a stimulus discretization, (iii) an empirical frequency function that measures the familiarity of stimuli, and (iv) a novelty function that computes the novelty of stimuli as a decreasing function of their familiarity [16–20, 22–24].

The stimulus representation specifies how stimuli from the environment are represented. For example, Gabor-like visual stimuli could be characterized by their angular orientation (Fig. 1 A, left). The stimulus discretization specifies how stimuli are counted. Count-based novelty models use separate counters for different stimuli. Every time a stimulus s is observed, its counter $C(s)$ is increased by one unit (Fig. 1 A, middle). Examples where stimuli can be clearly identified as 'distinct' and 'identical' in this sense are human studies in abstract sequential decision-making tasks [17,43]. In continuous environments and for stimuli with continuous features like the orientation of Gabor filters, however, two stimuli are neither fully 'distinct' nor 'identical' – instead, they show varying degrees of similarity. In these settings, counting is not well-defined. Even though we can manually construct a discretization by binning the stimulus space, e.g. into equidistant bins based on psychometric discrimination thresholds, binning still fundamentally misses the continuous nature of stimulus similarities, and fails to capture their impact on novelty computation (see toy example below).

The third element of count-based novelty models is the empirical frequency function p that measures the familiarity of a given state s at time t:

$$p^{(t)}(s) = \frac{C^{(t)}(s) + 1}{t + |S|}, \tag{1}$$

where $C^{(t)}(s)$ is the count of how often stimulus s has been observed up to the current time step t, and $|S|$ is the number of different stimuli. At time $t = 0$, all stimuli start with the same prior frequency $p^{(0)}(s) = 1/|S|$. The frequency increases if a stimulus s is observed (by increasing the count C and time t), and decreases if it is not observed (by increasing only the time t). In its most general form, count-based novelty allows the stimulus counts $C^{(t)}(s)$ to be 'leaky', i.e., decaying with a factor $(1 - \alpha)$ in each time step; $\alpha = 0$ corresponds to the non-leaky count-based novelty model. Some count-based models compute the state familiarity using simpler functions than the stimulus frequency, e.g., absolute counts [16,22,41,44] or binary counts [18,23]. Since these models can all be reformulated in terms of the empirical frequency p [29,36], we focus on frequency-based novelty models as in Eq. 1 for the rest of the manuscript.

The third element of count-based novelty models is the novelty function N. The novelty of a stimulus s decreases nonlinearly with its familiarity:

$$N^{(t)}(s) = -\log p^{(t)}(s), \tag{2}$$

where the negative logarithm is a standard choice of nonlinearity in the novelty literature [36, 42].

The combination of empirical count-based frequency (Eq. 1) and logarithm (Eq. 2) makes the novelty decrease rapidly during the first few observations of a stimulus s (Fig. 1 A, right), in line with experimental observations [39].

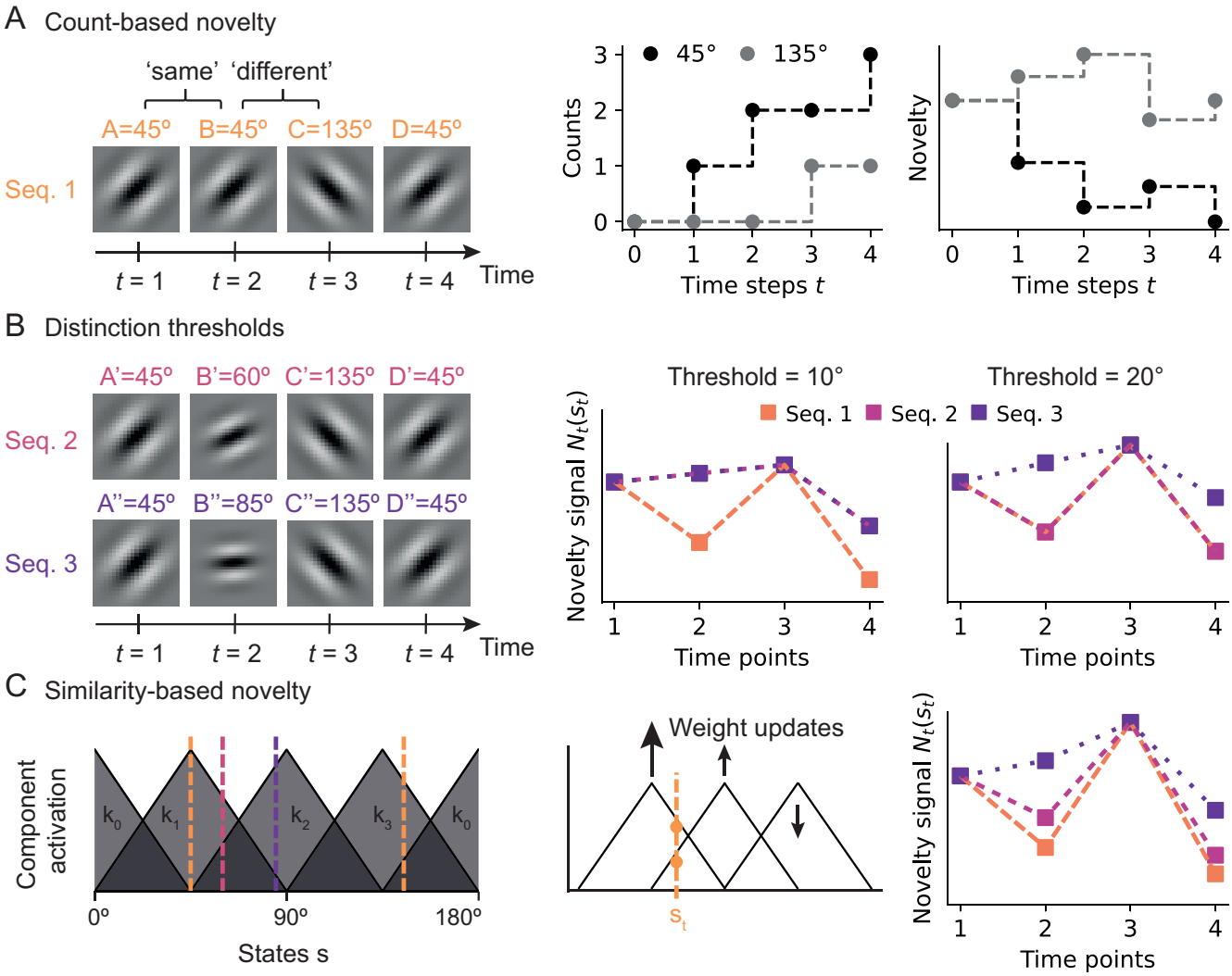

Figure 1: From count-based to similarity-based novelty. (A) Stimulus counts (middle) and count-based novelty responses (right) for an example sequence of four Gabors, characterized by their angular orientations between 0° and 180° (left). Each time we observe a stimulus, its count increases by one unit. Different stimuli (e.g. here: 45° vs. 135°) are counted separately. Stimulus novelty is a decreasing function of the stimulus count. **(B)** Novelty responses for two count-based novelty models with different stimulus discretization. (left) Two additional example sequences of four Gabors each. The three sequences in (A) and (B) are identical except for the second stimulus, which shows varying level of similarity to the first stimulus in each sequence. (middle, right) Novelty predictions for the three Gabor sequences using two count-based novelty models. The models count stimuli as 'different' if they differ by more than 10° (middle) or 20° (right), respectively. The two count-based models make different predictions. **(C)** Similiary-based novelty using triangular components centered at reference orientations 0°, 45°, 90° and 135° (left). Component weights encode the empirical frequency of each component; they increase relative to the activation of their associated component (middle). Similarity-based novelty predictions (right) reflect the different levels of similarity between the first two stimuli across the three sequences: The novelty predicted for the second stimulus increases from seq. 1 to seq. 3, as the similarity between the first two stimuli decreases.

To illustrate the behavior of count-based novelty models, we consider a toy example: We present three stimulus sequences with varying levels of stimulus similarities to two count-based novelty models with different state discretizations (Fig. 1 A-B, left). Each sequence consists of four Gabor stimuli that are each characterized by their angular orientation. The three sequences are identical except for the second stimulus, which has varying levels of similarity to the first (seq. 1: identical,

seq. 2: 15° difference, seq. 3: 40° difference). We evaluate the novelty response to each stimulus as predicted by two count-based novelty models: The first model considers Gabors as distinct if they differ by more than 10° (Fig. 1 B, middle); the second one if they differ by more than 20° (Fig. 1 B, right). The two threshold values of 10° and 20° are consistent with psychometric thresholds for orientation differences measured in mice [45]. For the first sequence (yellow line in Fig. 1 B) and the third sequence (purple line in Fig. 1 B), both count-based novelty models predict qualitatively similar novelty signals: Every time we observe the first stimulus, its novelty decreases (seq. 1: at $t = 2, t = 4$, seq. 3: at $t = 4$), while previously unseen stimuli show high novelty responses (seq. 1: $t = 1, t = 3$, seq. 3: $t = 1, t = 2, t = 3$). Since we are measuring frequencies, and the initial counts are zero for all stimuli, the responses to completely novel stimuli increase during the observation of the sequence (with different slopes, depending on the discretization, i.e. the number of 'bins'). Importantly, for the second sequence (pink line in Fig. 1 B), the two count-based models predict different novelty responses. The more fine-grained discretization (Fig. 1 B, middle) predicts the same novelty responses for seq. 2 and seq. 1 ('identical' first two stimuli), while the more coarse-grained discretization (Fig. 1 B, right) predicts the same responses to seq. 2 and seq. 3 ('distinct' first two stimuli). This exemplifies a fundamental limitation of count-based novelty models: They do not account for stimulus similarities beyond a simple distinction between 'same' and 'different'. In naturalistic stimulus spaces, where stimuli vary continuously and stimulus similarities are abundant, these model predictions contradict both common-sense intuition and experimental evidence [13, 26–28] (see Discussion).

Similarity-based novelty

To address the limitations of count-based novelty models, we propose a generalized novelty model ('similarity-based novelty') that accounts for the effect of stimulus similarities and allows to model novelty in continuous stimulus spaces. Its core difference to count-based novelty models is that, instead of computing the familiarity of stimuli as a function of their *discrete* empirical frequency (Eq. 1), our model uses probabilistic mixture models to compute a *continuous* empirical frequency density across the stimulus space S.

The idea of mixture models is to represent a complex probability density as the weighted sum of simpler probability densities (mixture 'components') over the stimulus space (see e.g. [46]). The mixture components can have arbitrary shapes, as long as they are non-negative and their integral over the stimulus space is one. In general, each component can overlap with an arbitrary number of other components. In the context of our problem, mixture components can serve as a flexible way to express stimulus similarities in both discrete and continuous spaces. For example, in our toy example, we can choose overlapping triangular components to account for orientation similarities (Fig. 1 C). Each component encodes the similarity of any given stimulus to a fixed reference stimulus, i.e., the orientation at the center of the triangular component. The pairwise similarity between two arbitrary stimuli can thus be described by the overlap in components they activate.

Based on the components, we define a mixture model p that generalizes the empirical frequency in Eq. 1. It quantifies the relative familiarity of a stimulus s at time t as

$$p^{(t)}(s) = \sum_{j=1}^{N} w_j^{(t)} k_j(s), \tag{3}$$

where $k_1, ..., k_N$ are the components that account for the similarity structure of the stimulus space, and $w_1^{(t)}, ..., w_N^{(t)}$, with $\sum_{j=1}^{N} w_j^{(t)} = 1$, are the weights with which each component contributes to

the empirical frequency density $p^{(t)}$ at a given time point t. In order for the empirical frequency density p to represent a meaningful and consistent notion of familiarity across space and time, we must choose the weights $w_j^{(t)}$ appropriately. We can do so using statistical inference: After a sequence of observations $s_1, ..., s_t$, the ideal weights should be chosen so as to maximize the likelihood of the observed sequence. This normative approach leads to an iterative update rule:

$$w_j^{(t)} = w_j^{(t-1)} + \alpha^{(t)} \delta_j^{(t)}, \tag{4}$$

where $\alpha^{(t)} = \frac{1}{t+N\epsilon}$ (see Methods) is the time-dependent learning rate, and the error $\delta_j^{(t)}$ for the weight of component k_j at time t is

$$\delta_j^{(t)} = \left(\frac{k_j(s_t)}{p^{(t-1)}(s_t)} - 1 \right) w_j^{(t-1)}. \tag{5}$$

Crucially, the error in Eq. 5 depends on the previous weights through a multiplication factor, i.e., the familiarity weights are updated with a *multiplicative* learning rule, similar to what has been suggested in a range of mechanistic models of novelty detection [26,37,38,47–49] (see Discussion). Since the error in Eq. 5 only relies on information that is locally available in time, i.e. the current component activation $k_j(s_t)$ and the current familiarity $p^{(t-1)}(s_t)$, it can be implemented in a biologically plausible neural circuit model (see Methods and SI). Analogously to the 'leaky' count-based novelty, we can define 'leaky' weight updates by choosing a constant weight update rate α in Eq. 4, resulting in an approximate maximum likelihood estimate of the component weights.

Using the learning rule in Eq. 4 along with suitable stimulus and component representations, we estimate the familiarity of a stimulus s at time t by its empirical frequency $p^{(t)}(s)$. Note that the definition of familiarity in Eq. 3-4 implies that similar stimuli influence each other's familiarity via the components they share. For example, if a stimulus is covered by two (overlapping) components, its familiarity will be influenced by other stimuli that share any of the two mixture components. Analogously to count-based novelty (Eq. 2), we then apply the negative logarithm to $p^{(t)}(s)$ to compute the (continuous) novelty $N^{(t)}(s)$. The resulting generalized novelty model ('similarity-based novelty') is (i) well-defined in discrete and continuous stimulus spaces, (ii) can account for similarities between stimuli via its component representation, and (iii) is equivalent to the established count-based novelty in discrete stimulus spaces without similarity structure (see SI).

To illustrate the differences between count-based and similarity-based novelty, we compare their novelty predictions in our toy example (Fig. 1 A-B, left). Analogous to the count-based model (Fig. 1 B, middle and right), we predict the novelty responses to each stimulus in each sequence using similarity-based novelty (Fig. 1 C). As components, we choose equidistantly placed, triangular functions that encode orientation similarities as linearly decreasing function of their angular difference (Fig. 1 C, left and middle). This resembles a linear approximation to the tuning curves of orientation-selective cells in V1 [50] (see [51] for a review). The predictions of similarity-based novelty are qualitatively different from the predictions of count-based novelty. In contrast to count-based novelty, similarity-based novelty predictions for the second stimulus increase with the angular difference to the first stimulus (Fig. 1 C, right), reflecting the varying degree of similarity between the first two stimuli across the three sequences. An analogous similarity-driven decrease of novelty responses is predicted for the stimulus at time $t = 4$. This shows that, in contrast to count-based novelty, similarity-based novelty generalizes familiarity across stimuli proportionally to their similarity, as defined by the underlying component representation.

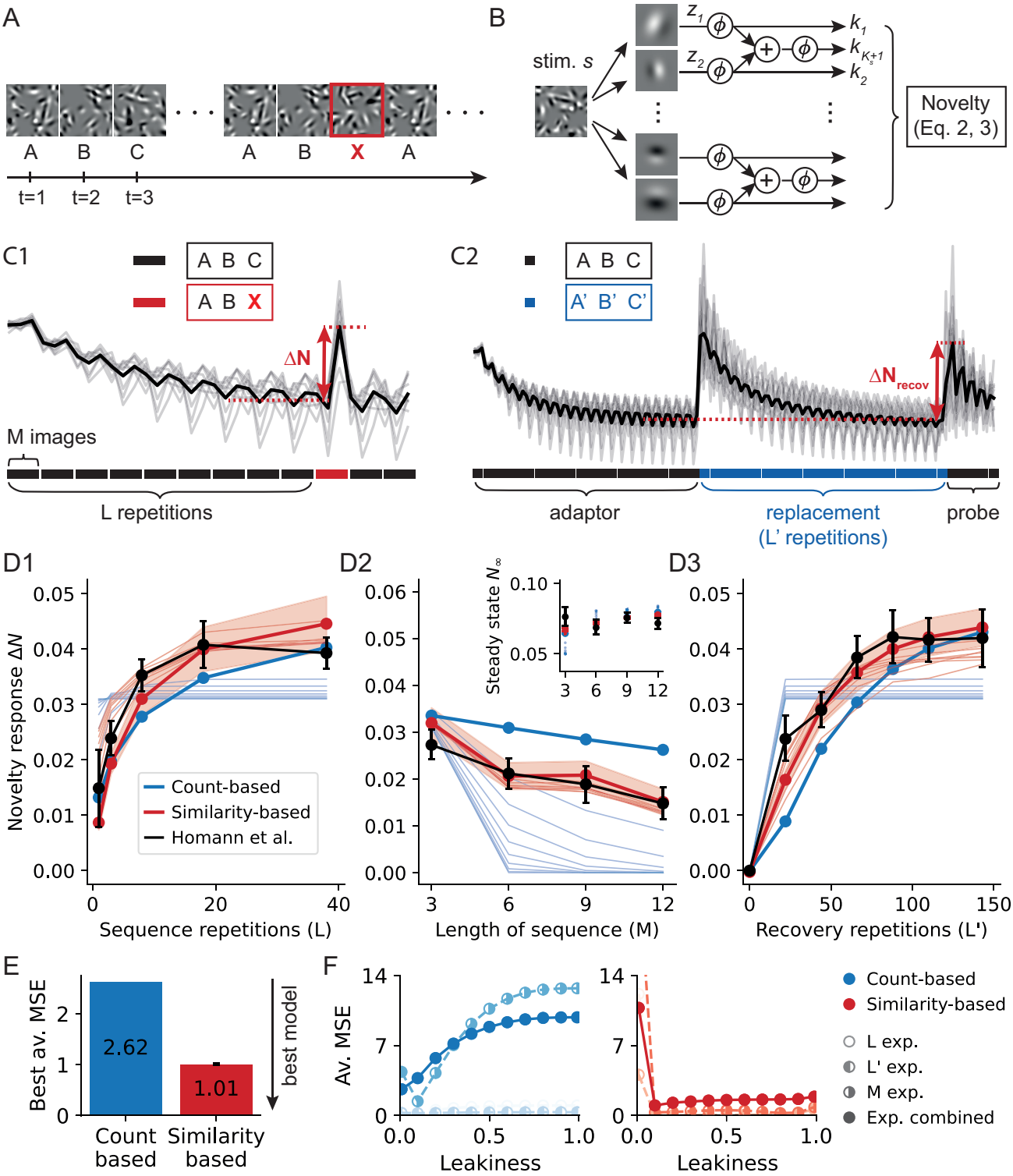

Figure 2: Similarity-based novelty robustly explains novelty responses in mouse V1. (caption on next page)

Figure 2: **(A)** Passive viewing task by Homann et al. [38]. A sequence of images is presented repeatedly, until the last image in the sequence is replaced by a novel, i.e. previously unseen, image (X). Familiar images in the sequence and novel images X are drawn from the same distribution and each consist of a linear superposition of 40 Gabor filters with random parameters. **(B)** The components of similarity-based novelty (k_j in Eq. 3) emulate V1 simple and complex cells [50, 51]. 'Simple cell'-like components compute the cosine similarity between the stimulus image s and their reference Gabor (defined by its preferred orientation, phase, frequency, and location) and pass it through an arbitrary non-negative function ϕ (see Methods, here: ReLU, same for all components) to compute their component activation. The activation of a 'complex cells'-like component is computed by applying the same function ϕ to the sum of 2-4 'simple cell'-like component activations with opposite phase and different frequencies. **(C)** Three variations of the passive viewing experiment in A. We consider responses to the 'novel' image for (i) different numbers L of sequence repetitions ($M = 3$ fixed; C1) and (ii) different numbers M of 'familiar' images in a sequence ($L = 18$ fixed; C1). We compare the novelty signals ΔN predicted by count-based and similarity-based novelty to the population responses of neurons in mouse V1 [38]. We further assess the recovery from familiarity in a third experiment (C2): after novelty responses to an initial sequence ('adaptor') have converged, a second, different sequence ('replacement') is presented for L' repetitions; subsequently, the recovered novelty response ΔN_{recov} to the initial sequence ('probe') is measured ($M = 3$ fixed). **(D)** V1 novelty responses ΔN [38] (black line) and novelty predictions ΔN of fitted (best av. MSE) count-based (bold blue line) and similarity-based novelty models (bold red line) in all three experiments. For the M-experiment (D2), responses to the 'novel' image (ΔN) and steady state response to the 'familiar' sequence (N_∞, inset) are shown. Thin blue and red lines depict novelty predictions of each novelty model for the remaining (non-optimal) values of the leakiness parameter α in F. Similarity-based novelty reproduces all qualitative features of neural novelty responses in V1, robustly for different values of α, whereas count-based novelty is highly susceptible to changes in α. **(E)** Average mean squared error (MSE) between neural novelty responses [38] and novelty responses predicted by count-based and similarity-based novelty, across the three experiments. **(F)** Average MSE between model predictions and neural responses across experiments as a function of the leakiness level α for count-based novelty (left, blue) and similarity-based novelty (right, red). Count-based novelty shows a bad fit (large av. MSE) everywhere except in a small region around leakiness $\alpha = 0.1$ due to large errors in the M-experiment. Similarity-based novelty gives a robustly good fit (low av. MSE) across all leakiness levels (=novelty learning rates) except in the limit of very fast learning ($\alpha \to 0$). Average MSE is stable across other model parameters (see SI).

Similarity-based novelty explains novelty responses in mouse V1

In the previous section, we showed how we can model the impact of stimulus similarities on novelty computation using our similarity-based novelty model. In this section, we ask whether similarity-based novelty can capture key features of novelty responses in the brain, and provide further insights into how stimulus similarities modulate novelty computation in sensory areas. To this end, we compare the novelty signals predicted by count-based and similarity-based novelty with neural novelty responses in the mouse primary visual cortex (V1) during passive viewing of a sequence of 'familiar' and 'novel' images, as reported by Homann et al. [38].

The visual stimuli in the experiment of Homann et al. [38] are images that each consist of multiple randomly placed and randomly oriented Gabor filters (Fig. 2 A). Homann et al. study V1 novelty responses in mice during three variations of their passive viewing experiment (Fig. 2 C1-C2), each illustrating a specific qualitative feature of novelty responses in V1. In the first version of the experiment, a sequence of three images ($M = 3$) is presented for L repetitions ($L = 1, 3, 8, 18, 38$) before a 'novel' image (X) is presented in the subsequent repetition (Fig. 2 C1). Here, a 'novel' image is defined as an image that is not in the set of 'familiar' images that are shown as part of the repeating image sequence. The neural population response to the novel image increases with

L (Fig. 2 D1, black line). In the second variation of the experiment, the number M of familiar images in the sequence is varied ($M = 3, 6, 9, 12$; Fig. 2 C1), while the number of sequence repetitions L remains constant ($L = 18$). Increasing M decreases the neural responses to the novel image (Fig. 2 D2, black line) but does not significantly affect the steady-state population activity after habituation to the familiar sequence (Fig. 2 D2, black line in inset). The third variation investigates the recovery from familiarity: after neural responses to a familiar sequence ('adaptor', $M = 3$) have converged, a novel sequence of images ('replacement', $M = 3$) is presented for L' repetitions, before the initial sequence is shown again and the neural population response ΔN_{recov} is measured (Fig. 2 C2). The recovered novelty response to the formerly familiar sequence increases with L' (Fig. 2 D3, black line).

To test whether these qualitative features of novelty responses in V1 can be captured by count-based and similarity-based novelty, respectively, we fit each novelty model to the combined experimental data [38] from all three experimental variations (see Methods). The best-fitting parameters for each model are determined by minimizing the mean squared error (MSE) between the statistics of predicted and the experimentally measured neural novelty responses, averaged across the three experiments. For the count-based novelty model, all images are considered as distinct. For the similarity-based novelty model, we define components that emulate either V1 simple cells with specific orientation, phase, frequency and location preference, or V1 complex cells that combine the feature preferences of at least two V1 simple cells with opposite phase (Fig. 2 B, see Methods for details). The number of bins (for count-based novelty) and components (for similarity-based novelty) are free parameters that are fitted jointly with the remaining parameters of each model using grid search (see Methods).

In all three experiments, similarity-based novelty captures V1 novelty responses better than count-based novelty (Fig. 2 D1-D3 and E). Additionally, similarity-based novelty maintains its good fit across the majority of the parameter space, while the fit of count-based novelty is highly susceptible to parameter changes (Fig. 2 F, see SI for robustness results for remaining parameters). In particular, the fit of count-based novelty decreases rapidly with deviations from the fitted leakiness level α (Fig. 2 F, left panel). Similarity-based novelty captures the neural data well across the majority of leakiness levels α (Fig. 2 F, right panel). Only in the limit of very low leakiness levels α, the fit of similarity-based novelty decreases rapidly. This is to be expected since, for similarity-based novelty, the 'leakiness' level α is equivalent to the learning rate with which observations are integrated into the component weights; very small values of α thus prevent new observations from being incorporated into the familiarity estimate. The fit is stable with respect to the remaining model parameters for both novelty models (see SI). To further test the validity of our fit, we compare the statistics of novelty predicted by the fitted count-based and similarity-based model, respectively, with the statistics of neural novelty responses in V1 (Fig. 2 D1-D3). As expected, the novelty predicted by the similarity-based novelty model (red line in Fig. 2 D1-D3) captures the features of V1 novelty responses in all three experiments, robustly for fitted parameters (thick red line) and deviating leakiness levels (i.e., learning rates) α (thin red lines). In contrast, count-based novelty shows deviations from the V1 novelty responses in all three experiments (thick blue line) that rapidly increase as we deviate from the fitted leakiness level α (thin blue lines). Specifically, non-optimal count leakiness leads to instant saturation of the novelty responses predicted by count-based novelty in all three experiments (Fig. 2 D1-D3, thin blue lines).

Taken together, these findings suggest that count-based and similarity-based novelty reproduce V1 novelty responses using two qualitatively different mechanisms. Count-based trades off the timescales required for each experiment, resulting in high susceptibility to the leakiness parameter. For example, a higher than optimal count leakiness in the L-experiment causes the predicted novelty responses to reach their steady state in very few repetitions, preventing a gradual saturation of novelty responses as observed in the neural data. At the same time, a slower than

optimal count-learning rate prevents the decline of novelty responses with increasing stimulus sequence length M. In contrast, similarity-based novelty reproduces V1 novelty responses using a largely generalization-based mechanism. For example, in the M-experiment, similarity-based generalization of familiarity leads to a natural decrease of novelty responses as the number M of familiar stimuli and, thus, the feature overlap with the novel stimulus increases. As a results, similarity-based novelty avoids a trade-off of learning timescales and ultimately achieves a better and more robust fit to neural responses across all three experiments.

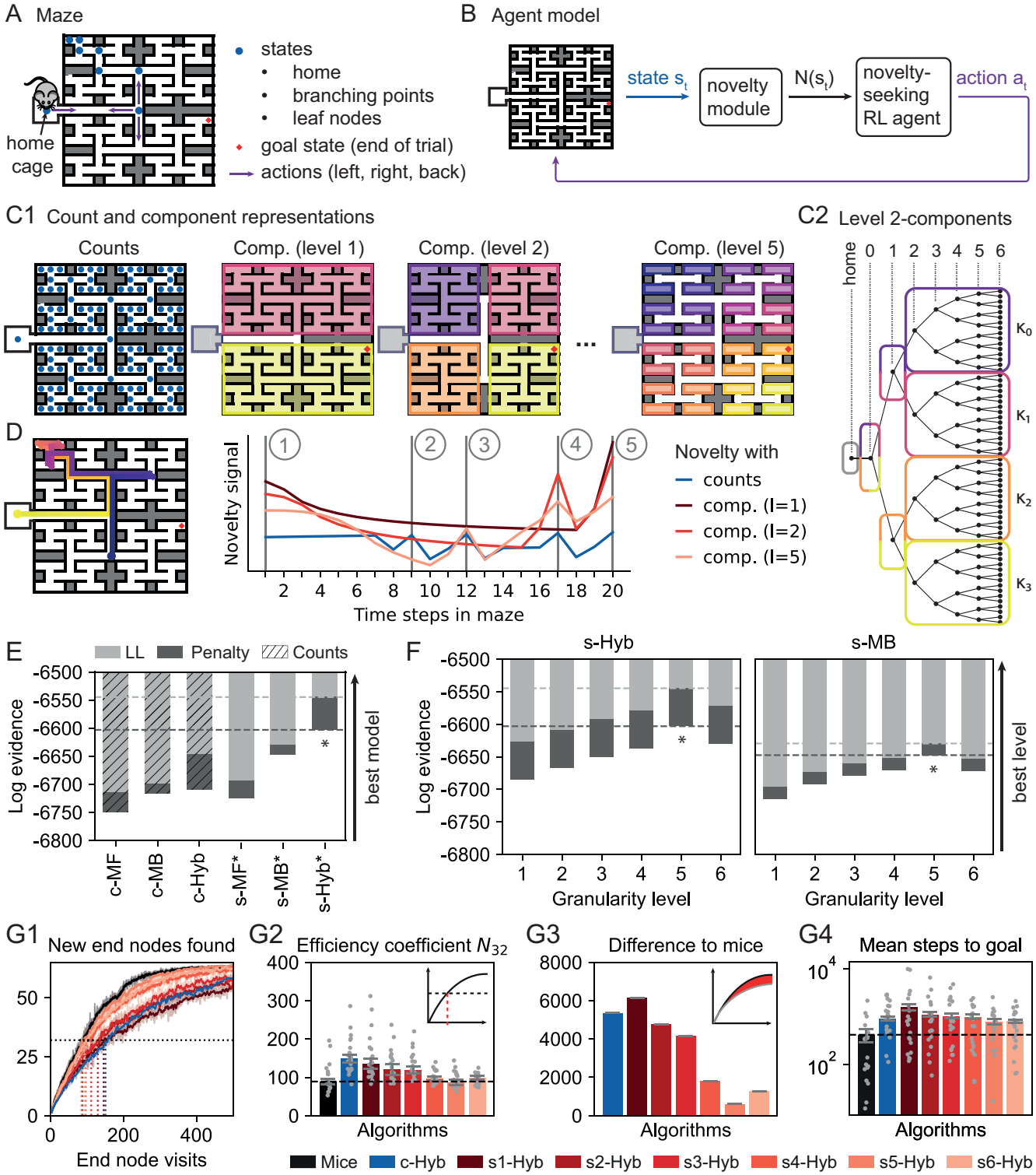

Figure 3: Similarity-based novelty explains mouse exploration in an unfamiliar maze. (caption on next page)

Figure 3: Similarity-based novelty explains mouse exploration in an unfamiliar maze. (A) The maze of Rosenberg et al. [52]. We define states (blue circles, example states on a short trajectory) with respective actions (purple arrows): (i) a single action in the home cage (go into the maze), (ii) three actions at 'branching points' (turn left, right or backwards), and (iii) one action in the 'leaf nodes' (turn back). To investigate intrinsically driven exploration, we consider mouse behavior before their first encounter with the rewarded 'goal' (red square). **(B)** In each time step t, a novelty-seeking RL agent takes its current position (state s_t) as input and computes the state novelty using its internal novelty model. Then it updates its internal value function and policy based on a given RL algorithm (either model-free, model-based or hybrid), and chooses its next action a_t from its policy. **(C1-C2)** Counts and components used for the RL agents' internal novelty models. (C1, first panel) State counts for count-based novelty. (C1, remaining panels) Components that tile the state space into 'areas' of the maze, e.g., into quadrants in the 'level 2' representation. Each component is active in a given area of the maze (marked in different colors) and in the states along the shortest path between the home cage and its encoding area. For example, the purple 'level 2' component k_0 is active in all states within the purple quadrant and the two branching point states that connect it to the home cage (C2). **(D)** Novelty signals predicted by count-based and similarity-based novelty with different component granularity on an example path through the maze. Numbers indicate relevant behavioral events along the path: first entry into the maze (1), into novel level-6 component (2), level-5 component (3), level-2 component (4) and level-1 component (5). **(E)** Model fit of different novelty-seeking RL models to mouse exploration behavior. Log-evidence (loglikelihood augmented by penalty for number of parameters) of count-based (prefix: c-) and similarity-based (prefix: s-) novelty-seeking RL agents (model-free: MF, model-based: MB, hybrid: Hyb). For similarity-based novelty agents, the best model log-evidence across six granularity levels is reported (asterisk). Following the established interpretation scale for log-evidence [53, 54], a log-evidence difference of more than 3 is considered significant. **(F)** Comparison of similarity-based novelty with different granularity for the winning algorithm families (hybrid, model-based). Granularity 5 explains data best (asterisk). The best model overall is the hybrid RL agent with similarity-based novelty of granularity 5. **(G1-G4)** Posterior predictive checks of the model comparison in A-B. (G1) mean number of new leaf nodes found, as a function of total leaf node visits. Dashed vertical lines indicate the 'search efficiency' N_{32}, i.e. the number of leaf node visits until 32 (half) of all 64 leaf nodes have been visited. (G2) Search efficiency N_{32}); (G3) integrated difference between the end node discovery curves of agents and mice in G1; and (G4) mean number of steps until the first goal visit. Error bars indicate the standard error of the mean (sem).

Similarity-based novelty explains mouse exploration in an unfamiliar maze

In the previous section, we showed that similarity-based novelty explains key features of novelty responses in mouse V1 during a passive viewing task [38]. In this section, we ask whether similarity-based novelty also helps explaining behavioral signatures of novelty processing during active exploration. To this end, we analyze behavioral data by Rosenberg et al. [52] who measure exploration of mice in an unfamiliar, freely accessible maze (Fig. 3 A). The maze has the structure of a binary tree (Fig. 3 C2): Mice enter the maze through a single corridor that then branches into two corridors, which each branch again into two corridors etc. After six branching points, each corridor ends in a wall ('leaf node'), where mice can only turn around. Half of the mice do not receive any reward, the other half receive a drop of water upon licking a reward port in one of the maze's leaf nodes ('goal state'). To make sure that we only include novelty-driven behavior in our analysis, we consider mice behavior before the first encounter with the goal in both the unrewarded and the rewarded group. During this initial exploration phase, mice from the rewarded and the unrewarded group have access to identical information and do not show significant behavioral differences [52].

We model mouse behavior using novelty-seeking reinforcement learning (RL) models that max-

imize intrinsically computed novelty instead of extrinsic rewards from the environment (Fig. 3 B, see Methods). In order to compare the exploration behavior of different 'agents', i.e. mice and novelty-seeking RL models, we define 'states' and 'actions' in the labyrinth. States include the home cage, the branching points and the leaf nodes (blue circles in Fig. 3 A). In each of the branching points, mice or RL agents can take one of three actions (purple arrows in Fig. 3 A): (i) going down the left corridor until the next branching point or leaf node, (ii) going down the right corridor until the next branching point or leaf node, or (iii) going back to the previous branching point. In the home cage and the leaf nodes, agents only have one available action, i.e. going back.

We compare mouse behavior to three families of novelty-seeking RL models, i.e., model-free (MF), model-based (MB) and hybrid (Hyb) RL algorithms (see Methods), and ask whether similarity-based novelty explains mouse behavior better than count-based novelty. For count-based novelty, we count each state in the maze separately. For similarity-based novelty, we define components that encode an abstract notion of spatial proximity in the binary tree by considering (i) how the mice got from the home cage to their current location (shortest path between a state s and the home state), and (ii) which 'area' of the maze the mice are in (e.g., a given side or a given quadrant of the labyrinth). The size of the 'area' encoded by a given component defines the spatial scale ('granularity level') across which it generalizes familiarity. For example, a 'level 2' component (Fig. 3 C1-C2) generalizes across all states in one quadrant and the shortest path leading to it from the home cage, while a 'level 5' component (Fig. 3 C1) generalizes across the two closest branching points of a given leaf node and the corresponding shortest path from the home cage. To test how this spatial scale of components affects novelty-driven exploration, we compare seven similarity-based novelty models with components of different granularity levels (levels 0-6, see Methods). Count-based and similarity-based novelty models with different granularity predict different novelty signals along an example path through the maze (Fig. 3 D).

To test whether exploration driven by similarity-based novelty helps explaining mouse behavior, we fit all of the model variants described above, i.e. all combinations of RL models (MF, MB, Hyb) and novelty models (count-based, similarity-based with different granularity) to mouse behavior, using maximum likelihood estimation of the parameters (see Methods). To determine which model family explains the data best, we then perform a Bayesian model comparison between the best-performing models of each family (see Methods). Across all types of RL models, exploration driven by similarity-based novelty gives a significantly better fit to mouse behavior than exploration driven by count-based novelty (Fig. 3 E). Model comparison between model-based and hybrid similarity-based agents shows that components of granularity 5, which generalize across two immediately neighboring leaf nodes and their shared branching point, best explain the exploration behavior of mice (Fig. 3 F).

To further validate our model comparison results, we perform posterior predictive checks [55] (see Methods). In particular, we simulate count-based and the different similarity-based novelty-driven RL models and compare them to mice, with respect to the efficiency of their exploration (Fig. 3 G). As expected, the winning model (similarity-based novelty-driven model with level-5 granularity) approximates mouse behavior best and shows the most efficient exploration efficiency among all tested models (Fig. 3 G). In contrast, both the absence of generalization (count-based novelty) and over-generalization (similarity-based novelty with broad components) show significantly lower exploration efficiency than mice and the winning model. Taken together, these findings suggest that spatial similarity structures significantly modulate exploration of mice in unfamiliar environments via their influence on novelty signals.

Modeling novelty computation across naturalistic tasks

In this work, we propose a unified modeling framework for novelty ('similarity-based novelty') that generalizes the notion of novelty to continuous stimulus spaces and captures the effect of stimulus similarities on novelty (Fig. 4, left box). Its two main elements are (i) the set of non-negative component functions k_j that encode stimulus similarities in the representational space, and (ii) the history- and time-dependent component weights $w_j^{(t)}$ that encode the frequency of each component and are updated with a bio-plausible multiplicative learning rule (Eq. 4, 5). The familiarity of a given stimulus s is computed as the weighted sum of all component activations in response to s (Eq. 3), and the novelty of s is the negative log of its familiarity (Eq. 2). Since the weight update rule is directly derived from the normative approach and guarantees the frequency character of the familiarity density p, it remains the same across different choices of component sets. The component functions, on the other hand, can be flexibly adapted to model different data sets. While the similarity-based novelty framework allows arbitrary *non-negative* component functions, most component functions relevant for modeling consist of three elements: (i) a set of reference stimuli r_j that form 'representative' samples for similarity computation in the representational space of the input stimuli; (ii) a similarity measure f that is defined on the representational space; and (iii) a non-negative transform function ϕ that converts similarities into positive component activations.

In our first toy example, we chose the reference stimuli r_j to be equidistantly spaced orientations in the 1-dimensional space of Gabor orientations, the similarity f to be measured as Euclidean distance of angles, and the transform function ϕ to be a triangular function that can be thought of as a simple approximation of Gaussian receptive fields (Fig. 4, right upper box). To model novelty for the more complex visual stimuli of Homann et al.'s experiment [38], we choose reference stimuli r_j that emulate the preferred features of simple and complex cells in V1, cosine similarity as a similarity measure f in the pixel space, and the same triangular transform function ϕ as in the toy example (Fig. 4, right middle box). Importantly, the reference stimuli in this application are defined in the same space as the input stimuli (pixel space) but – in contrast to the toy example – are not drawn from the same distribution as the input stimuli. To capture novelty in the binary tree maze of Rosenberg et al. [52], we defined branching points of the maze as reference stimuli r_j, a shortest path-based spatial distance measure f, and the same triangular transform ϕ as in the previous two data sets (Fig. 4, right lower box). This exemplifies both the flexibility of the similarity measure and the possibility to combine our novelty framework with intrinsically motivated RL to model behavior. Thus, with a single probabilistic framework for novelty, we can model novelty computation, and its modulation by stimulus similarities, across passive and active tasks and behavioral and neural data sets.

Our findings suggest that stimulus similarities modulate novelty signals and novelty-driven behaviors in mice. However, while several studies [16, 21, 24, 38] test the impact of recency and repetition on the novelty of a stimulus, we still lack systematic studies about how stimulus similarities impact novelty computation. Using similarity-based novelty, we provide testable experimental predictions in a novel experimental protocol ('sum-of-parts' experiment, Fig. 4 C1-C3). Like in the M-experiment of Homann et al. [38], a sequence of familiar images consisting of M images ($M = 3, 6, 9, 12$) is presented for L repetitions, before a novel image X is presented. In contrast to the previous experiments, however, familiar stimuli are chosen such that they make up equal parts of the 'novel' image X (Fig. 4 C1). To vary the degree of similarity between familiar sequence and the image X, a controlled transform is applied to X (here: rotation of the Gabors in the image X). Importantly, the stimuli in the familiar sequence are fully distinct (zero pairwise similarity) but each of them has equal similarity to the novel image X (Fig. 4 C2, upper panel). The higher the rotation of the novel image X, the lower the similarity between familiar images in the sequence and X (Fig. 4 C2, lower panel). The same experiment could also be performed, e.g., with more

naturalistic images as long as they preserve the similarity relationships between familiar and novel images. Using similarity-based novelty, we predict the novelty responses for different lengths M of the familiar sequence ($M = 3, 6, 9, 12$) and for different levels of similarity between the familiar and novel images (Fig. 4 C3). Similarity-based novelty predicts that, for high similarity between familiar and novel images, novelty responses decrease with M due to similarity-based generalization of familiarity. For low similarity between familiar and novel images, we predict the *opposite* trend, i.e. that novelty responses increase with M due to the increased number of total stimulus presentations in longer sequences. Our model simulations suggest that, as the similarity between familiar and novel image increases, the novelty responses gradually shift from the 'generalization'-dominant regime (high similarity, dark blue in Fig. 4 C3) to the 'recency'-dominant regime (low similarity, light blue in Fig. 4 C3). This example protocol illustrates how similarity-based novelty enables theory-driven experiment design.

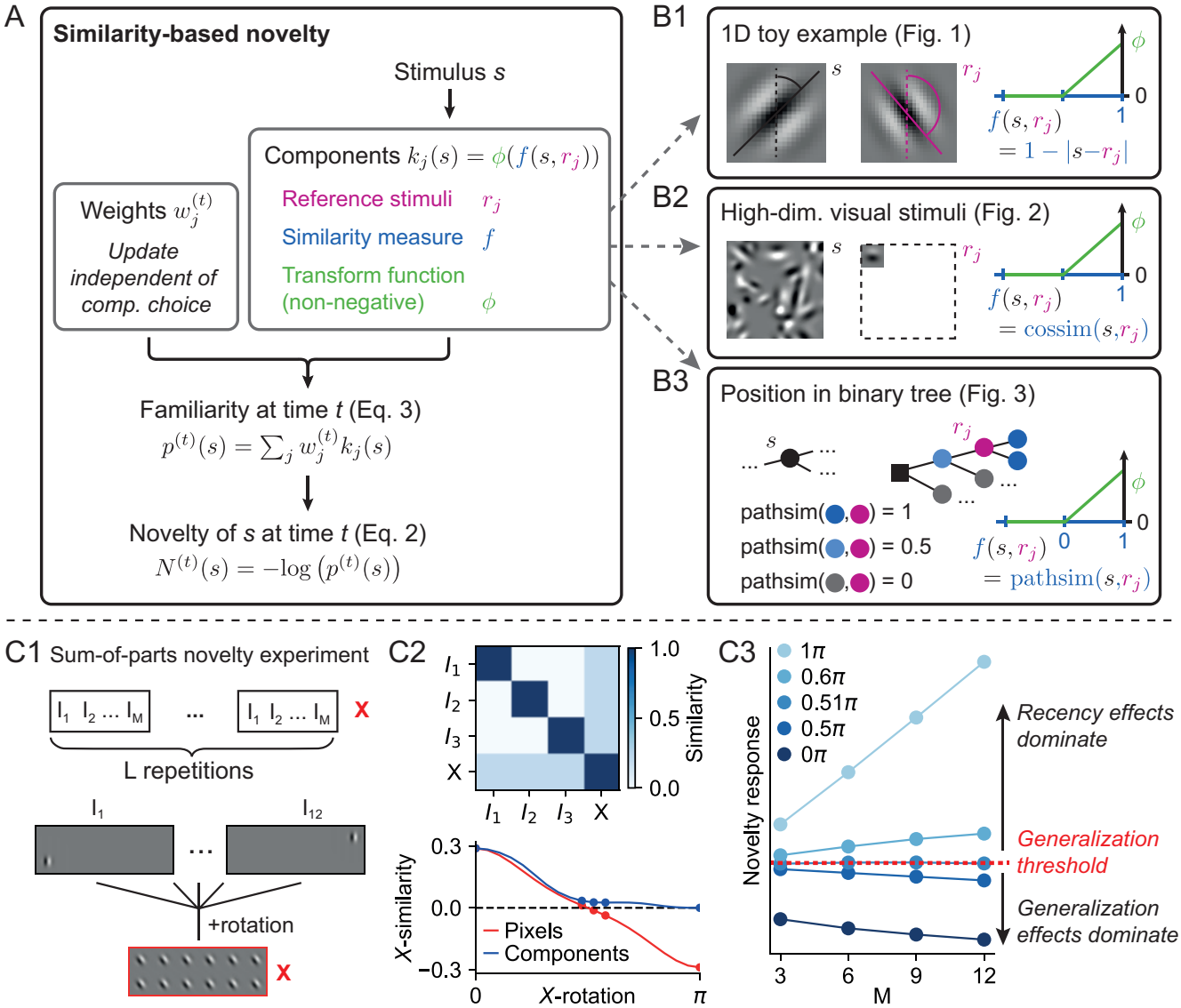

Figure 4: Modeling novelty computation across naturalistic tasks. (A) The similarity-based novelty framework represents the familiarity of stimuli as a probabilistic mixture model, using (i) non-negative component functions that capture similarities between stimuli, and (ii) component weights encoding the empirical frequency of each component in past observations. **(B1-B3)** The similarity-based novelty framework can be adapted to modeling neural data and behavior across different naturalistic tasks, enabling task- and stimulus-specific novelty predictions with a single probabilistic framework. **(C1-C3)** Schematic of a theory-inspired experimental protocol to further investigate the effects of stimulus similarities on novelty computation. (C1) A sequence of M familiar images ($M = 3, 6, 9, 12$) is presented for L repetitions, before a 'novel' image X is presented. Familiar stimuli are chosen such that they make up equal parts of the 'novel' image X. The degree of similarity between familiar sequence and the image X is varied by rotating the Gabors in X compared to the Gabors in the familiar sequence. (C2) Equal pairwise similarity between images in the familiar sequence (here: 0 similarity), and between each familiar image and X (upper panel). Higher rotation of X leads to lower pairwise similarity between any familiar image and X. Circles denote the rotation values for which novelty predictions were computed in C3. (C3) Novelty signals as predicted by similarity-based novelty for different values of M and different angles of rotation of X (i.e. different similarity between familiar and novel images). For high similarity (low rotation), novelty responses decrease with M (dark blue, 'generalization'-dominant regime); for low similarity (high rotation), novelty responses increase with M (light blue, 'recency'-dominant regime).

Discussion

We developed a generalized model of novelty computation that leverages mixture models [46] to capture the effect of stimulus similarities on novelty computation, and extends current computational novelty models to continuous state spaces. In tasks where all stimuli are discrete and distinct, our similarity-based novelty model is equivalent to count-based novelty [16–19,22–24,41,44]. We showed that similarity-based novelty captures novelty responses in mouse V1 [38] better and more robustly than count-based models. By combining similarity-based novelty with intrinsically motivated RL models [17, 22, 25, 29, 56], we further showed that seeking similarity-based novelty explains the exploration behavior of mice [52] better than seeking count-based novelty. Lastly, we use our similarity-based novelty model to make testable predictions in a novel experimental protocol.

Our results suggest that representational similarities between unfamiliar and familiar stimuli significantly influence both neural and behavioral signatures of novelty-related processing in the brain. Specifically, (i) similarity to familiar stimuli attenuates the novelty response to a previously unobserved stimulus, and (ii) generalization of familiarity across nearby spatial locations makes exploration more efficient by reducing the time spent exploring 'similar' (=close-by) regions in the maze. Our model unifies these findings by showing that they can both arise due to the modulation of novelty signals by stimulus similarities. Building on this, we predict that also novelty responses in other sensory modalities [4, 10, 27, 57] and in the hippocampus [6, 9, 58] as well as downstream novelty signaling in salience-related regions [5, 20, 21, 28, 59] could be subject to similarity modulation. If this prediction is true, it would have important implications for the experimental study of novelty: 'novel' input stimuli should either be chosen as almost perfectly distinct (e.g. [7,17,28]) to eliminate the impact of stimulus similarities on novelty signals, or their similarity should be controlled for as an experimental variable. Apart from enabling theory-driven experimental hypotheses and experiment design, our similarity-based novelty model also allows us to reinterpret existing experimental findings in the context of similarity-driven generalization of familiarity. For example, a study by Montgomery [13] investigating the impact of luminance on mice exploration in otherwise identical mazes finds that a larger luminosity difference (i.e. less similarity) between two successively explored mazes leads to more exploration in the second maze than a small or no luminosity difference (i.e. high similarity). This relationship is a signature of exploration driven by similarity-modulated novelty. Along the same lines, DeBaene and Vogels [26] report that neuronal adaptation in the monkey IT cortex significantly depends on the difference between adaptor and test stimulus, with respect to features such as object shape and location. This links directly to possible circuit implementations through which stimulus similarities could modulate novelty signals.

Moreover, insights from machine learning support the functional need to generalize novelty information across similar stimuli [25]. In particular, Jaegle et al. [25] argue that any system that computes novelty has to solve the problems of (i) mapping a diversity of high-dimensional inputs to reasonable states that abstract away e.g. different views on the same object ('view invariance'), and (ii) grouping states together based on their shared characteristics ('state invariance'). Similarity-based novelty solves the second problem by expressing similarities in a given representation space by (potentially overlapping) component functions. In our work, we addressed the first challenge, i.e., the choice of an appropriate state representation, by constructing stimulus representations based on experimental knowledge and hypotheses. In general, however, similarity-based novelty is not limited to such 'experimentally justified' representations and can also be used, e.g., in the latent space of trained deep networks with 'brain-like' representations [60,61]. As such, similarity-based novelty extends computational models of novelty to naturalistic tasks, allowing them to be modeled in a unified normative framework. Some machine learning algorithms solve both view and state invariance problem simultaneously by training a deep network to estimate

a familiarity density from which novelty can be computed via pseudo-counts [29, 30]. Others use hashing [31] to map similar states to discrete bins that can then be used with count-based novelty (see [62] for an interesting neural implementation), or construct meaningful latent spaces where the novelty of a state is defined as a function of how many other states are within a given Euclidean distance in the latent space [34]. The fact that most of these methods are trained 'end-to-end' is both an advantage, since they learn an appropriate state representation simultaneously with the familiarity density, and a disadvantage, since they require learning of a separate density estimation network using backpropagation instead of computing novelty from existing representations in a modular fashion. In particular, they are not compatible with hand-designed representations based on 'experimental knowledge', making it harder to investigate, e.g., how a certain distribution of place fields influence novelty-driven exploration, or how novelty signals in V1 depend on the specific tuning of their cells.

One of the central advantages of similarity-based novelty is its biologically interpretable (local) learning rule. Specifically, the weights that capture the empirical frequency of each component are updated with a multiplicative learning rule. While similarity-based novelty does not make explicit predictions about which circuit structure has to underlie novelty computation, its update rule for the novelty weights is consistent with several types of circuit mechanisms proposed for novelty computation, including input adaptation and short-term synaptic depression [26, 37, 38, 47], inhibitory circuits [40], or Hebbian/anti-Hebbian multiplicative plasticity [48, 49]. As such, similarity-based novelty proposes a normative perspective on a wide range of mechanistic novelty models that provide area- and modality-specific mechanistic implementations of novelty computation.

While novelty has been shown to be a dominant drive of human exploration in unfamiliar environments [17, 19], a diversity of other intrinsic reward-like signals are involved in driving exploration [9, 13–15, 17–19, 21, 63, 64] (see [59, 65–68] for reviews), especially when extrinsic rewards such as food and money are absent. The relative importance of different motivational signals in explaining behavior depends on multiple factors, including the temporal stages of exploration (see e.g. [17]) as well as the exploration objectives and features of the task and environment (see e.g. [69]). Our findings for novelty suggest that stimulus similarities might be another aspect that substantially influences the computation of these intrinsic signals and their impact on behavior. In neuroscience, however, most of these motivational signals are investigated in a count-based setting. The question how stimulus similarities impact the computation of these diverse set of motivational signals, including surprise and information gain, thus remains an important direction for future research.

In conclusion, we propose a unifying computational framework to model novelty computation in the brain ('similarity-based novelty') that allows to assess the impact of similarity-driven generalization on neural novelty responses and novelty-driven behavior. Our approach opens new possibilities for future theory-driven experiment design.

References

[1] Gregor Rainer and Earl K. Miller. Timecourse of object-related neural activity in the primate prefrontal cortex during a short-term memory task. *European Journal of Neuroscience*, 15:1244–1254, 2002.

[2] John E. Lisman and Anthony A. Grace. The hippocampal-VTA loop: controlling the entry of information into long-term memory. *Neuron*, 46:703–713, 2005.

[3] Tomonori Takeuchi, Adrian J. Duszkiewicz, Alex Sonneborn, Patrick A. Spooner, Miwako Yamasaki, Masahiko Watanabe, Caroline C. Smith, Guillén. Fernández, Karl Deisseroth, Robert W. Greene, and Richard G. M. Morris. Locus coeruleus and dopaminergic consolidation of everyday memory. *Nature*, 537:357–362, 2016.

[4] Joachim Morrens, Çağatay Aydin, Aliza Janse van Rensburg, José Esquivelzeta Rabell, and Sebastian Haesler. Cue-evoked dopamine promotes conditioned responding during learning. *Neuron*, 106(1):142 – 153.e7, 2020.

[5] Munir G. Kutlu, Jennifer E. Zachry, Patrick R. Melugin, et al. Dopamine signaling in the nucleus accumbens core mediates latent inhibition. *Nature Neuroscience*, 25:1071–1081, 2022.

[6] James B. Priestley, John C. Bowler, Sebi V. Rolotti, Stefano Fusi, and Attila Losonczy. Signatures of rapid plasticity in hippocampal CA1 representations during novel experiences. *Neuron*, 110:1978–1992, 2022.

[7] A. Ghazizadeh and O. Hikosaka. Salience memories formed by value, novelty and aversiveness jointly shape object responses in the prefrontal cortex and basal ganglia. *Nature Communications*, 13, 2022.

[8] Nico Bunzeck and Emrah Düzel. Absolute coding of stimulus novelty in the human substantia nigra/VTA. *Neuron*, 51:369–379, 2006.

[9] Emrah Düzel, Nico Bunzeck, Marc Guitart-Masip, and Sandra Düzel. Novelty-related motivation of anticipation and exploration by dopamine (NOMAD): implications for healthy aging. *Neuroscience & Biobehavionral Reviews*, 34:660–669, 2010.

[10] Nicholas C. Foley, David C. Jangraw, Christopher Peck, and Jacqueline Gottlieb. Novelty enhances visual salience independently of reward in the parietal lobe. *Journal of Neuroscience*, 34(23):7947–7957, 2014.

[11] Ali Ghazizadeh, Whitney Griggs, and Okihide Hikosaka. Ecological origins of object salience: Reward, uncertainty, aversiveness, and novelty. *Frontiers in Neuroscience*, 10, 2016.

[12] Daniel Ernst, Stefanie Becker, and Gernot Horstmann. Novelty competes with saliency for attention. *Vision Research*, 168:42–52, 2020.

[13] K. C. Montgomery. Exploratory behavior as a function of 'similarity' of stimulus situations. *Journal of Comparative and Physiological Psychology*, 46:129–133, 1953.

[14] Daniel E. Berlyne. Novelty and curiosity as determinants of exploratory behaviour. *British Journal of Psychology. General section*, 41:68–80, 1950.

[15] Daniel E. Berlyne. Curiosity and exploration. *Science*, 153:25–33, 1966.

[16] Rachit Dubey and Thomas L Griffiths. Reconciling novelty and complexity through a rational analysis of curiosity. *Psychological Review*, 127(3):455–476, 2020.

[17] He A. Xu, Alireza Modirshanechi, Marco P. Lehmann, Wulfram Gerstner, and Michal H. Herzog. Novelty is not Surprise: Human exploratory and adaptive behavior in sequential decision-making. *PLoS Computational Biology*, 17:e1009070, 2021.

[18] Irene C. Dezza, Xavier Noel, Axel Cleeremans, and Angela J. Yu. Distinct motivations to seek out information in healthy individuals and problem gamblers. *Translational Psychiatry*, 11:408, 2021.

[19] Alireza Modirshanechi, He A. Xu, Wei-Hsiang Lin, Michael H. Herzog, and Wulfram Gerstner. The curse of optimism: a persistent distraction by novelty. *bioRxiv*, 2022.

[20] Akiti Korleki, Iku Tsutsui-Kimura, Yudi Xie, Alexander Mathis, Jeffrey E. Markowitz, Rockwell Anyoha, Sandeep Robert Datta, Mackenzie Weygandt Mathis, Naoshige Uchida, and Mitsuko Watabe-Uchida. Striatal dopamine explains novelty-induced behavioral dynamics and individual variability in threat prediction. *Neuron*, 110:3789–3804, 2022.

[21] Takaya Ogasawara, Fatih Sogukpinar, Kaining Zhang, Yang-Yang Feng, Julia Pai, Ahmad Jezzini, and Ilya E. Monosov. A primate temporal cortex–zona incerta pathway for novelty seeking. *Nature Neuroscience*, 25:50–60, 2022.

[22] Samuel J. Gershman and Yael Niv. Novelty and inductive generalization in human reinforcement learning. *Topics in cognitive science*, 7(3):391–415, 2015.

[23] Samuel J Gershman, Marie-H Monfils, Kenneth A Norman, and Yael Niv. The computational nature of memory modification. *Elife*, 6:e23763, 2017.

[24] Maxime Maheu, Stanislas Dehaene, and Florent Meyniel. Brain signatures of a multiscale process of sequence learning in humans. *Elife*, 8:e41541, 2019.

[25] Andrew Jaegle, Vahid Mehrpour, and Nicole Rust. Visual novelty, curiosity, and intrinsic reward in machine learning and the brain. *Current Opinion in Neurobiology*, 58:167–174, 2019.

[26] Wouter De Baene and Rufin Vogels. Effects of adaptation on the stimulus selectivity of macaque inferior temporal spiking activity and local field potentials. *Cerebral Cortex*, 20:2145–2165, 2009.

[27] Travis Meyer and Nicole C. Rust. Single-exposure visual memory judgments are reflected in inferotemporal cortex. *Elife*, 7:e32259, 2018.

[28] Kaining Zhang, Ethan S. Bromberg-Martin, Fatih Sogukpinar, Kim Kocher, and Ilya E. Monosov. Surprise and recency in novelty detection in the primate brain. *Current Biology*, 32(10):2160–2173.e6, 2022.

[29] Marc Bellemare, Sriram Srinivasan, Georg Ostrovski, Tom Schaul, David Saxton, and Remi Munos. Unifying count-based exploration and intrinsic motivation. In *Advances in Neural Information Processing Systems*, pages 1471–1479, 2016.

[30] Georg Ostrovski, Marc G. Bellemare, Aäron van den Oord, and Rémi Munos. Count-based exploration with neural density models. In Doina Precup and Yee Whye Teh, editors, *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pages 2721–2730. PMLR, 06–11 Aug 2017.

[31] Haoran Tang, Rein Houthooft, Davis Foote, Adam Stooke, Xi Chen, Yan Duan, John Schulman, Filip De Turck, and Pieter Abbeel. #Exploration: A Study of Count-Based Exploration for Deep Reinforcement Learning. In *Advances in Neural Information Processing Systems*, volume 30, 2017.

[32] Yuri Burda, Harrison Edwards, Amos Storkey, and Oleg Klimov. Exploration by random network distillation. *arxiv*, 2018.

[33] Nikolay Savinov, Anton Raichuk, Raphaël Marinier, Damien Vincent, Marc Pollefeys, Timothy Lillicrap, and Sylvain Gelly. Episodic curiosity through reachability. *ICLR*, 2019.

[34] Ruo Yu Tao, Vincent François-Lavet, and Joelle Pineau. Novelty search in representational space for sample efficient exploration. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 8114–8126. Curran Associates, Inc., 2020.

[35] Marlos C. Machado, Marc G. Bellemare, and Michael Bowling. Count-based exploration with the successor representation. *Proceedings of the AAAI Conference on Artificial Intelligence*, 34:5125–5133, 2020.

[36] Alireza Modirshanechi, Sophia Becker, Johanni Brea, and Wulfram Gerstner. Surprise and novelty in the brain. *Current Opinion in Neurobiology*, 82:102758, 2023.

[37] Robert Mill, Martin Coath, Thomas Wennekers, and Susan L. Denham. A neurocomputational model of stimulus-specfic adaptation to oddball and markov sequences. *PLoS Computational Biology*, 7:e1002117, 2011.

[38] Jan Homann, Sue A. Koay, Kevin S. Chen, David W. Tank, and Michael J. Berry. Novel stimuli evoke excess activity in the mouse primary visual cortex. *Proceedings of the National Academy of Sciences*, 119:e2108882119, 2022.

[39] Rafal Bogacz and Malcolm W. Brown. Comparison of computational models of familiarity discrimination in the perirhinal cortex. *Hippocampus*, 13:494–524, 2003.

[40] Auguste Schulz, Christoph Miehl, Michael J. Berry II, and Julijana Gjorgjieva. The generation of cortical novelty responses through inhibitory plasticity. *Elife*, 10:e65309, 2021.

[41] Sham Kakade and Peter Dayan. Dopamine: generalization and bonuses. *Neural Networks*, 15(4-6):549–559, 2002.

[42] Andrew Barto, Marco Mirolli, and Gianluca Baldassarre. Novelty or surprise? *Frontiers in psychology*, 4:907, 2013.

[43] Nathaniel D Daw, Samuel J Gershman, Ben Seymour, Peter Dayan, and Raymond J Dolan. Model-based influences on humans' choices and striatal prediction errors. *Neuron*, 69(6):1204–1215, 2011.

[44] Alexander L. Strehl and Michael L. Littman. An analysis of model-based interval estimation for Markov decision processes. *J Comput Syst Sci*, 74:1309–1331, 2008.

[45] Lupeng Wang, Kerry McAlonan, Sheridan Goldstein, Charles R. Gerfen, and Richard J. Krauzlis. A causal role for mouse superior colliculus in visual perceptual decision-making. *Journal of Neuroscience*, 40:3768–3782, 2020.

[46] Christopher M. Bishop and Nasser M. Nasrabadi. *Pattern recognition and machine learning*, volume 4. Springer, 2006.

[47] Kyle Aitken, Luke Campagnola, Marina Garrett, Shawn Olsen, and Stefan Mihalas. Familiarity modulated synapses model visual cortical circuit novelty responses. *bioRxiv*, 2023.

[48] Sukbin Lim, Jillian L. McKee, Luke Woloszyn, Yali Amit, David J. Freedman, David L. Sheinberg, and Nicolas Brunel. Inferring learning rules from distributions of firing rates in cortical neurons. *Nature Neuroscience*, 18:1804–1810, 2015.

[49] Krithika Mohan, Ulises Pereira-Obilinovic, Stanislav Srednyak, Yali Amit, Nicolas Brunel, and David Freedman. Visual familiarity learning at multiple timescales in the primate inferotemporal cortex. *bioRxiv*, 2024.

[50] D. H. Hubel and T. N. Wiesel. Receptive fields, binocular interaction and functional architecture in the cat's visual cortex. *Journal of Physiology*, 160:106–154, 1962.

[51] Nicholas J. Priebe1. Mechanisms of orientation selectivity in the primary visual cortex. *Annual Review of Vision Science*, 2:85–107, 2016.

[52] Matthew Rosenberg, Tony Zhang, Pietro Perona, and Markus Meister. Mice in a labyrinth show rapid learning, sudden insight, and efficient exploration. *eLife*, 10:e66175, 2021.

[53] Robert E Kass and Adrian E Raftery. Bayes factors. *Journal of the american statistical association*, 90(430):773–795, 1995.

[54] Harold Jeffreys. *The theory of probability*. OuP Oxford, 1998.

[55] Andrew Gelman, John B. Carlin, Hal S. Stern, David B. Dunson, Aki Vehtari, and Donald B. Rubin. *Bayesian Data Analysis*. Chapman and Hall/CRC, 3 edition, 2013.

[56] Nuttapong Chentanez, Andrew G Barto, and Satinder P Singh. Intrinsically motivated reinforcement learning. In *Advances in neural information processing systems*, pages 1281–1288, 2005.

[57] Itai Hershenhoren, Nevo Taaseh, Flora M. Antunes, and Israel Nelken. Intracellular correlates of stimulus-specific adaptation. *The Journal of Neuroscience*, 34:3303–3319, 2014.

[58] Ruy Gomez-Ocadiz, Massimiliano Trippa, Chun-Lei Zhang, and et al. A synaptic signal for novelty processing in the hippocampus. *Nature Communications*, 13, 2022.

[59] Ilya E. Monosov. Curiosity: primate neural circuits for novelty and information seeking. *Nature Reviews Neuroscience*, 25, 2024.

[60] Dan Yamins and James J. DiCarlo. Using goal-driven deep learning models to understand sensory cortex. *Nature Neuroscience*, 19, 2016.

[61] Martin Schrimpf, Jonas Kubilius, Michael J. Lee, N. Apurva Ratan Murty, Robert Ajemian, and James J. DiCarlo. Integrative benchmarking to advance neurally mechanistic models of human intelligence. *Neuron*, 108, 2020.

[62] Sanjoy Dasgupta, Daisuke Hattori, and Saket Navlakha. A neural theory for counting memories. *Nature Communications*, 13:5961, 2022.

[63] Franziska Brändle, Lena J. Stocks, Joshua B. Tenenbaum, Samuel J. Gershman, and Eric Schulz. Empowerment contributes to exploration behaviour in a creative video game. *Nature Human Behavior*, 7, 2023.

[64] Jorge Ramirez-Ruiz, Dmytro Grytskyy, Chiara Mastrogiuseppe, Yamen Habib, and Ruben Moreno-Bote1. Complex behavior from intrinsic motivation to occupy future action-state path space. *arxiv*, 2024.

[65] Jacqueline Gottlieb and Pierre-Yves Oudeyer. Towards a neuroscience of active sampling and curiosity. *Nature Reviews Neuroscience*, 19(12):758–770, 2018.

[66] Rachit Dubey and Thomas L. Griffiths. Understanding exploration in humans and machines by formalizing the function of curiosity. *Current Opinion in Behavioral Sciences*, 35:118–124, 2020.

[67] Mousa Karayanni and Israel Nelken. Extrinsic rewards, intrinsic rewards, and non-optimal behavior. *Journal of Computational Neuroscience*, 50:139–143, 2022.

[68] Alireza Modirshanechi1, Kacper Kondrakiewicz, Wulfram Gerstner, and Sebastian Haesler. Curiosity-driven exploration: foundations in neuroscience and computational modeling. *Trends in Neurosciences*, 46:1054–1066, 2023.

[69] Lucas Gruaz, Alireza Modirshanechi, Sophia Becker, and Johanni Brea. Merits of curiosity: a simulation study. *PsyArXiv*, 2024.

[70] S. J. Nowlan. *Soft Competitive Adaptation: Neural Network Learning Algorithms based on Fitting Statistical Mixtures*. PhD thesis, School of Computer Science, Carnegie Mellon University, Pittsburgh, 1991.

[71] Geoffrey J. McLachlan and Thriyambakam Krishnan. *The EM Algorithm and Extensions*. 2007.

[72] Cristopher M. Niell and Michael P. Stryker. Highly selective receptive fields in mouse visual cortex. *Journal of Neuroscience*, 28:7520–7536, 2008.

[73] Matthew Rosenberg, Tony Zhang, Pietro Perona, and Markus Meister. Rosenberg-2021-repository. `https://github.com/markusmeister/Rosenberg-2021-Repository`, 2021.

[74] Richard S Sutton and Andrew G Barto. *Reinforcement learning: An introduction*. MIT press, 2018.

Methods

Leaky updates for count-based novelty

In its most general form, count-based novelty (Eq. 1, 2) allows for 'leaky' updates of the stimulus counts $C^{(t)}(s)$:

$$C^{(t+1)}(s) = (1-\alpha)C^{(t)} + \delta(s, s_t), \tag{6}$$

where α is the decay parameter, and δ is the Kronecker delta function (1 if $s = s_t$, 0 elsewhere). The larger α, the faster the counts decay over time. For $\alpha = 0$, we recover the 'non-leaky' classical count-based novelty model.

Normative update rule for similarity-based novelty

The central element of similarity-based novelty is the mixture model that defines the familiarity density p (Eq. 3). Its weights at time point t are chosen as the maximum-a-posteriori (MAP) estimate of the sequence $s_{1:t}$ of stimulus observations up to t:

$$\mathbf{w}^{(t)} = \arg\max_{\mathbf{w}} \log \Pr(s_{1:t}|\mathbf{w}) \Pr(\mathbf{w}), \tag{7}$$

where $\mathbf{w}^{(t)} = \left(w_1^{(t)}, ..., w_N^{(t)}\right)^{\mathrm{T}}$ is the vector of mixture weights at time t, and $\Pr(\mathbf{w})$ is a Dirichlet prior over the weights (see SI). We solve Eq. 7 using the incremental expectation-maximization (EM) algorithm [46, 70, 71]. The resulting closed-form solution for the weights can then be reformulated as an online learning rule given in the main text (Eq. 4). The two weight update rules (batch and online) derived from the EM algorithms are given as follows.

(i) EM update (batch). For each time point t at which we observe a stimulus s_t, we compute the E-step (estimation step) of the incremental EM algorithm, i.e. we compute 'responsibilities' $\gamma_{j,t}$ that estimate how well a given component k_j of the mixture distribution at time $t-1$ captures the observation s_t:

$$\gamma_{j,t} = \frac{w_j^{(t-1)} k_j(s_t)}{\sum_{j'=1}^{N} w_{j'}^{(t-1)} k_{j'}(s_t)}. \tag{8}$$

The initial weights are chosen uniformly as $w_j^{(0)} = \frac{1}{N}$ for all $j = 1, ..., N$. In the subsequent M-step (maximization step) for time point t, we compute the new weights $w_j^{(t)}$ based on the responsibilities $\gamma_{j,t'}$, $j = 1, ..., N$, $t' = 1, ..., t$ that were estimated *in all previous* E-steps (Eq. 8):

$$w_j^{(t)} = \frac{1}{t + \epsilon(N-1)} \sum_{t'=1}^{t} \gamma_{j,t'}, \tag{9}$$

where ϵ is the uniform concentration parameter of the Dirichlet prior over the weights. The main difference to textbook incremental EM algorithms [46] is that at each iteration, the amount of data in the batch increases since an additional data point has been added to the sequence $s_1 : t$ (see SI for details).

(ii) Online update. The expression for the weights at time t (Eq. 9) still depend on the information from previous E-steps through the 'responsibilities' $\gamma_{j,t'}$, $t' = 1, ..., t-1$. However, we

can reformulate Eq. 9 to obtain an online weight update rule that computes $\mathbf{w}^{(t)}$ only based on the responsibilities from the current E-step at time t and the previous weights $\mathbf{w}^{(t-1)}$:

$$\begin{aligned} w_j^{(t)} &= \frac{1}{t+\epsilon(N-1)} \sum_{t'=1}^{t} \gamma_{j,t'} = \frac{1}{t+\epsilon(N-1)} \left(\gamma_{j,t} + \sum_{t'=1}^{t-1} \gamma_{j,t'} \right) \\ &= w_j^{(t-1)} + \frac{1}{t+\epsilon(N-1)} \left(\gamma_{j,t} - w_j^{(t-1)} \right), \end{aligned} \tag{10}$$

for all $j = 1, ..., N$. The learning rule in Eq. 10 can be written as a delta learning rule as described in the main text (Eq. 4):

$$w_j^{(t)} = w_j^{(t-1)} + \alpha^{(t)} \delta_j^{(t)}, \tag{11}$$

with error

$$\delta_j^{(t)} = \gamma_{j,t} - w_j^{(t-1)} = \left(\frac{k_j(s_t)}{p^{(t-1)}(s_t)} - 1 \right) w_j^{(t-1)} \tag{12}$$

and either time-dependent learning rate (derived from incremental EM)

$$\alpha^{(t)} = \frac{1}{t+\epsilon(N-1)} \tag{13}$$

or constant learning rate ('approximate' incremental EM). A constant learning rate yields a 'leaky' weight update, analogously to the 'leaky' count-based novelty. Our learning rule in Eq. 11 is *multiplicative* since the error (Eq. 12) depends on the previous weights $w_j^{(t-1)}$ through a multiplication factor. The multiplication factor of the weights in Eq. 10 depends on the ratio of component activation and familiarity of the current observation s_t, which are both available locally in time. For example, the component activation by the current observation s_t can be interpreted as the presynaptic activity of weight $w_j^{(t-1)}$, while the familiarity of s_t can be seen as the postsynaptic activity of the novelty-detecting circuit (see SI for a biologically plausible circuit implementation).

Toy experiment: Gabors sequences of different similarity

Novelty predictions of count-based and one similarity-based novelty models were computed for three sequences of four Gabor stimuli each. Gabor stimuli are characterized by their angular orientation. Across all sequences $i = 1, 2, 3$, all except the second stimulus s_2 are identical. The second stimulus in each sequence shows varying levels of similarity to the first stimulus: $s_2^{(1)} = 45°$ (identical to $s_1^{(1)}$), $s_2^{(2)} = 60°$ (15° difference to $s_1^{(2)}$), $s_2^{(3)} = 85°$ (40° difference to $s_1^{(3)}$).

Countable states for the count-based novelty models are obtained by binning the space of Gabor orientations $[0, 180°]$ into 9 bins of width 20° with centers $0°, 20°, 40°$, etc. (model variant 1) or into four bins of 45° with centers $0°, 45°, 90°$ and 135° (model variant 2). The size of the state space, $|S|$, is equal to the number of bins, respectively; counts at $t = 0$ were initialized at zero, and leakiness was set to $\alpha = 1$ (non-leaky).

The component representation of similarity-based novelty is defined on the 1D torus of Gabor orientations $[0, 180°]$. We use four equidistantly placed components k_j, $j = 1, ..., 4$, centered at $c_1 = 0°$, $c_2 = 45°$, $c_3 = 90°$ and $c_4 = 135°$. The components are defined by the following triangular function:

$$k_j(s) = \frac{1}{\sigma_j} \left(1 - \frac{|s - c_j|}{\sigma_j} \right) \Theta \left(1 - \frac{|s - c_j|}{\sigma_j} \right), \tag{14}$$

where $s \in [0°, 180°]$ is the orientation of the Gabor stimulus, c_j as defined above are the centers

of the components, and σ_j are the component widths. We choose $\sigma_j = 90°$ for all components, creating the overlapping components depicted in Fig. 1 C. The component weights at $t = 0$ were initialized uniformly to $\frac{1}{N}$, where $N = 4$ is the number of components. For the toy example simulations, we used non-leaky similarity-based novelty, with prior $\epsilon = 1$.

Passive viewing task: Experimental data by Homann et al.

Setup and recordings. Homann et al. [38] used two-photon calcium imaging to record the neural activity of layer 2/3 neurons in the primary visual cortex (V1) of 5 GCaMP6f-expressing mice during passive viewing of 'familiar' and 'novel' visual stimuli. During the recording, mice were head-fixed but could run freely on an air suspended Styrofoam ball in front of a toroidal screen, onto which the visual stimuli were projected. Homann et al. extracted the peak neural responses (trial- and population-averaged, see SI for details) to the novel image (ΔN) and to the familiar sequence (steady-state activity N_∞) during the 'variable repetition experiment' (L-experiment) and the 'variable image number experiment' (M-experiment), as well as the transient population response (ΔN_{recov}) to the formerly familiar sequence after the recovery interval in the 'repeated image set experiment' (L'-experiment).

Passive viewing task: Model simulations and fitting

Stimuli. All images for our simulation of the passive viewing experiment were sampled from a distribution, designed to match that of Homann et al. [38]. Each image is the linear superposition of 40 Gabor filters with randomly chosen orientation (uniformly sampled from $[0, \pi]$), phase (unif. sampled from $[-\pi, \pi]$), frequency (unif. sampled from $[0.04, 0.08]$), and location (x and y coordinate sampled uniformly from $[-130, 130]$ and $[-20, 90]$, in units of visual degrees) on a grey background. The width of each Gabor filter was set to $\frac{1}{3f}$, where f is the spatial frequency of the Gabor filter.

Simulation protocol. Model simulations followed the same protocol as the mouse experiment by Homann et al. Each simulation step corresponds to 300 ms in the experiment, i.e. in each simulation step, a single stimulus is presented.

Component definition. The input stimuli for similarity-based novelty are 100×200 pixel images as defined above. Each image is represented by a pixel vector $s \in \mathbb{R}^{20,000}$ with entries specifying the intensity of each image pixel (grey-scale; the lower the pixel value, the darker the color; grey background has pixel value 0). We define two types of component functions that emulate the receptive fields of V1 simple and complex cells, respectively.

'Simple cell'-like components:
Analogously to the component centers in the toy example, each component function k_j is defined relative to a 'reference Gabor filter' $r_j \in \mathbb{R}^{20,000}$, i.e. the pixel vector of a single Gabor filter that is chosen equidistantly from the same parameter space (orientation, phase, frequency, $x-y$-location) as the Gabor filters that form the stimulus images. The value of component k_j with reference Gabor r_j in response to a stimulus image s is defined as

$$k_j(s) = \phi(f(s, r_j)), \tag{15}$$

where f is the cosine distance between the pixel vectors s and r_j:

$$z_j = f(s, r_j) = \frac{s \cdot r_j}{\|s\| \|r_j\|} \quad \in [-1, 1], \tag{16}$$

and ϕ is a non-negative transform function that maps the similarity value between the stimulus and the reference Gabor (in the interval $[-1, 1]$) to the component activation. We choose the function ϕ to be triangular with width parameter $\sigma_j = 1$ and centered at $z = 1$ (maximum similarity):

$$\phi^{\triangle}(z_j) = \begin{cases} 1 - \frac{1|z-1|}{\sigma_j} & \text{if } z_j \in [1 - \sigma_j, 1], \\ 0 & \text{otherwise.} \end{cases} \tag{17}$$

'Complex cell'-like components:
'Complex cell'-like components are defined relative to a reference set of two 'simple cell'-like kernels of opposite phase. The activations from the set of reference components are added, and then passed through the same function ϕ as for the 'simple cell'-like components. 'Complex cell'-like components can be thought of as encoding an OR relation between the reference Gabors of the 'simple cell'-like components of their reference set.

We define K_s 'simple cell'-like components and K_c 'complex cell'-like components. We keep the ratio of 3:1 of simple cell-like to complex cell-like components, in alignment with experimental results for layer 2/3 of V1 in mice [72], where Homann et al. recorded. The total number of components $N = K_s + K_c$, the width $\sigma_j = \sigma$ (shared across all components) and learning rate α that determines the speed of similarity-based novelty updates are free parameters that are fitted to the neural data by Homann et al.

Comparison with count-based novelty. The count-based model receives the same inputs but does not use receptive fields. Instead, separate counts are used for different images. The parameter α is fit to data.

Fit to neural data. To fit each novelty model to the neural data, we use grid search in combination with linear regression. We simulate a given novelty model $\mathcal{M} \in \{$count-based novelty, similarity-based novelty$\}$ across the full range of model parameters ($\theta_{\text{count}} = (|S|, \alpha)$, $\theta_{\text{similarity}} = (N, \sigma, \alpha)$, see SI); we then fit the simulated novelty responses to the neural data using linear regression (see SI). The parameter set that minimizes the normalized mean-squared error (MSE), averaged across the three experiments, is chosen as the best-fit parameter set for model $\mathcal{M}$.

Active exploration task

Task. Rosenberg et al. [52] track the behaviour of 20 mice (10 of them water-deprived) during free exploration of an unfamiliar maze (maze cleaned with ethanol between mice). Each mouse was video-recorded during its first 7 hours of access to the maze The maze was constructed as a 6-level binary tree, with branching points and end points of corridors as nodes. One single end node in the maze contained a water delivery port, which delivered a water reward to water-deprived mice ('rewarded group') but was inactive during the experiment with non-deprived mice ('unrewarded group'). The x-y-coordinates of the animal's nose in each video frame were used as the 'mouse position' for subsequent behavioral analysis (both by Rosenberg et al. and in our study, see SI for details).

Environment formalization. To enable direct comparison between the behavior of mice and reinforcement learning (RL) models, we formalize the labyrinth environment using 'states' and

'actions'. 'States' s characterize the possible locations of agents (mice or RL models) in the labyrinth maze, and 'actions' a characterize the possible next direction the agent can take from a given state. Analogously to Rosenberg et al. [52], the home cage and all branching points and end points of the labyrinths corridors (i.e. all nodes in the binary tree) were defined as 'states'. We define four different actions: (i) go forward into maze (available in the home cage); (ii) go back (available in all states except the home cage); (iii) turn and go into the left corridor (available in all branching points) and (iv) turn and go into the right corridor (available in all branching points). For simplicity of notation, our notion of left and right is based on the egocentric view of the agent during an outward path. Based on the definition of states in the maze, mouse behavior over time is discretized into 'time points', where each time point marks the transition to a new state in the maze.

Preprocessing. For each mouse in Rosenberg et al.'s data set (github: [73]), we extract the sequence of states and actions taken at each time point between their first entry into the maze and their first encounter with the goal state. To exclude any reward-related behaviors from the analysis, we disregard mouse behavior after the first encounter with the goal state, independently of whether the mouse received a reward in the goal state (rewarded group) or not (unrewarded group). Since there are no significant behavioral differences between rewarded and unrewarded mice during the time before the first reward encounter [52], we treat them as equivalent (see SI for details on outliers).

Novelty-seeking reinforcement learning (N-RL) models

We adapt classical reinforcement learning (RL) algorithms to model novelty-seeking ('novelty-seeking RL (N-RL) agents') by replacing the extrinsic reward signal by an intrinsically computed novelty signal (see e.g. [17, 22, 25, 29, 56] for similar approaches).

Like classical RL agents, N-RL agents move in an environment which is defined by states s, and actions a that allow the agent to transition between states. In general, the environment can be probabilistic in the sense that a given action a in a given state s can lead the agent into different states s'. The probabilities for each such state transition (s, a, s') are called transition probabilities and summarized in the transition matrix $p \in \mathbb{R}^{|S|\times|A|\times|S|}$, where $|S|$ is the size of the state space S, and $|A|$ is the size of the action space A. In our model, the environment is deterministic, i.e., the correct transition matrix contains only binary entries.

The main difference between novelty-seeking and classical RL algorithms is that, instead of maximizing the discounted future rewards, N-RL algorithms maximize the discounted future novelty:

$$G_{N,t} = \sum_{t'=t}^{T_{\max}} \gamma^{\,t'} N^{(t')}(s_{t'}), \tag{18}$$

where γ is the discount factor, and the novelty $N^{(t')}(s_{t'})$ is computed using any suitable novelty model, e.g. count-based or similarity-based novelty. Analogous to reward-based V-values and Q-values, N-RL algorithms compute 'novelty-based' V-values and Q-values:

$$V_N^{(t)}(s) = \mathbb{E}\left[\sum_{t'=1}^{\infty} \gamma^{\,t'} N^{(t')}(s_{t'}) | s_1 = s\right], \tag{19}$$

$$Q_N^{(t)}(s,a) = \mathbb{E}\left[\sum_{t'=1}^{\infty} \gamma^{\,t'} N^{(t')}(s_{t'}) | s_1 = s, a_1 = a\right]. \tag{20}$$

Active exploration task: Model simulations and fitting

We simulate and fit a total of 24 novelty-seeking RL (N-RL) models to the exploration behavior of the mice in the Rosenberg maze, and compare the fitted models with respect to their log-evidence. We consider models from three different RL model families: (i) model-based N-RL models based on Prioritized Sweeping [74] (Alg. 1 in SI), (ii) model-free N-RL models with Actor-Critic architecture [74] (Alg. 3 in SI), and (iii) hybrid N-RL models that combines the model-free and model-based variants using a hybrid action policy (Alg. 4 in SI). For each N-RL model family, we consider one model with count-based novelty, and seven models with similarity-based novelty, one for each of the component representations (see below).

Count-based novelty. Count-based novelty in the Rosenberg maze uses a separate count for each state as defined above, i.e. each node in the binary tree representation of the maze. State counts are updated in a non-leaky fashion ($\alpha = 1$). Using Eqs. 1 and 2, the novelty reward at each time is then computed as the novelty of the current state in the environment.

Similarity-based novelty. We define similarity-based novelty in the Rosenberg maze with respect to seven component representations that differ in the degree of granularity with which they encode the maze.

The component representations are based on the idea that each component should encode a different 'area' of the labyrinth environment, e.g. the left or right half of the maze (allocentric perspective), as well as a trace of how to reach this 'area' from the home cage. The size of the area encoded by the components determines the 'granularity' of the representation, i.e. the larger the area encoded by each component, the lower the granularity of the representation. To formally write the corresponding component functions, we consider the binary-tree structure of the labyrinth maze (Fig. 3C2) and define the '(sub-)tree' $\text{Tree}(s)$ for any state s in the labyrinth ($s \in S$) as

$$\text{Tree}(s) = \{s' \in S : s' \text{ is part of a binary tree with root state } s\}. \tag{21}$$

Hence, the entire labyrinth (excluding the home state) can be written as $\text{Tree}(s_0)$ where s_0 is the first branching point of the labyrinth. We further define the sets S_ℓ, $\ell = \{0, 1, ..., 6\}$, where each set S_ℓ contains all 2^ℓ states in a given level ℓ of the binary tree. For example, the set S_0 contains only s_0, the first branching point of the maze, while S_6 contains all 64 leaf nodes.

We define the component representation of granularity level ℓ, for $\ell = 1, ..., 6$ as follows. The component representation of a given granularity ℓ consists of $2^\ell + 1$ components: first, the home cage component, that is the same across all granularity levels and encodes the home cage separately:

$$k_H(s) = \begin{cases} 1 & \text{if } s = s_H \\ 0 & \text{otherwise;} \end{cases} \tag{22}$$

and second, the components $k_j^{(\ell)}$ that each encode the 'area' around one of the states $s_j^{(\ell)} \in S_\ell = \{s_1^{(\ell)}, ..., s_{2^\ell}^{(\ell)}\}$:

$$k_j^{(\ell)} = \begin{cases} \frac{1}{K^{(\ell)}} & \text{if } s \in \text{Tree}(s_j^{(\ell)}) & \text{('area')} \\ \frac{1}{2^{\ell-\ell'} K^{(\ell)}} & \text{if } s \in S_{\ell-\ell'} \text{ and } s_j^{(\ell)} \in \text{Tree}(s),\ \ell' = 0, ..., \ell & \text{('trace')} \\ 0 & \text{otherwise.} \end{cases} \tag{23}$$

The factor $K^{(\ell)} = K_j^{(\ell)} = \sum_{s \in S} k_j^{(\ell)}(s)$ is added to normalize the components to 1 over the state space.

Model fitting. We fit the parameters of each N-RL model to mouse behavior using maximum-likelihood estimation (MLE):

$$\hat{\theta} = \arg\max_{\theta} \mathrm{LL}(\mathcal{D}, \mathcal{M}, \theta) = \arg\max_{\theta}(\log P(\mathcal{D}|\mathcal{M}, \theta)), \tag{24}$$

where $\mathcal{M}$ is the N-RL model that is being fitted, θ are its model parameters, and $\mathcal{D}$ is the mouse data. The mouse data $\mathcal{D}$ consists of the appended state-action sequences of all mice during their exploration of the labyrinth:

$$\mathcal{D} = (s_0^{(1)}, a_0^{(1)}, \ldots, a_{T_1-1}^{(1)}, s_{T_1}^{(1)}, \quad \ldots \quad, s_0^{(20)}, a_0^{(20)}, \ldots, a_{T_{20}-1}^{(20)}, s_{T_{20}}^{(20)}), \tag{25}$$

where $s_t^{(i)}$ and $a_t^{(i)}$ denote the state and action of mouse i at time point t, and T_i denotes the number of exploration steps of mouse i until its first encounter of the goal state.

We maximize the log-likelihood for a given model by minimizing its negative log-likelihood using the scipy.optimize implementation of the Nelder-Mead algorithm (best performing among relevant optimization algorithms, including L-BFGS-B, SLSQP). In each minimization step, the log-likelihood for the current parameter set θ is computed (see SI), and its negative is used as input to the minimizer of the scipy.optimize package to compute the next candidate parameter set. We run the optimization until convergence to obtain the fitted parameters for a given model.

Bayesian model comparison. We compare fitted models with respect to their their log-evidence (LE):

$$\mathrm{LE}(\mathcal{D}, \mathcal{M}) = \mathrm{LL}(\mathcal{D}, \mathcal{M}, \theta)) - \frac{|\theta_{\mathcal{M}}|}{2} \log(T), \tag{26}$$

where the log-likelihood term accounts for the fit of the model, and the remaining term introduces a cost for model complexity, that is a function of the number of free parameters $|\theta_{\mathcal{M}}|$ of the model, and the number of data points $T = \sum_{i=1}^{20} T_i$ in the fitted data set $\mathcal{D}$.

Posterior predictive checks. We perform posterior predictive checks to verify that the winning model from our model comparison matches the fitted data (i.e. mouse behavior) with respect to relevant statistics of their behavior. To this end, we simulated all fitted models ($n = 500$ instantiations for each model). Each model simulation starts with the model agent being initialized in the home state, and ends when the model agent first reaches the goal state.

We compare mice and simulated models with respect to (i) the average number of steps in the maze until reaching the goal state; (ii) the average number of novel end nodes visited relative to the total number of visits to a (familiar or novel) end node; (iii) the average search efficiency N_{32}, i.e. the total number of end node visits until 32 (i.e. half) of the end nodes in the maze have been discovered; (iv) the integrated difference between the average end nodes discovery curve of mice and the average end node discovery curve of a given model. For the fourth statistic, the sem is computed using bootstrapping on the set of end node discovery curves of individual agents of a given type.

Supplementary Information

1 Novelty models

1.1 Normative approach for similarity-based novelty

As described in the main manuscript, we estimate the familiarity density $p^{(t)}$ at time point t,

$$p^{(t)}(s) = \sum_{j=1}^{N} w_j^{(t)} k_j(s), \tag{S.1}$$

by choosing the mixture weights at time t, $\mathbf{w}^{(t)} = \left(w_1^{(t)}, ..., w_N^{(t)}\right)^{\mathrm{T}}$, as the maximum-a-posteriori (MAP) estimate of the sequence $s_{1:t}$ of observations up to t:

$$\mathbf{w}^{(t)} = \arg\max_{\mathbf{w}} \log \Pr(s_{1:t}|\mathbf{w}) \Pr(\mathbf{w}), \tag{S.2}$$

where $\Pr(\mathbf{w})$ is a Dirichlet prior over the weights, i.e.

$$\Pr(\mathbf{w}) = c\left(\alpha\right) \prod_{j=1}^{N} (w_j)^{\alpha_j - 1}, \tag{S.3}$$

with concentration parameters $\alpha = (\alpha_1, ..., \alpha_N)^T$ and

$$c(\alpha) = \frac{\Gamma\left(\sum_{j=1}^{N} \alpha_j\right)}{\prod_{j=1}^{N} \Gamma\left(\alpha_j\right)}. \tag{S.4}$$

To compute the MAP estimate of the weights, we use the EM algorithm which allows us to compute the MAP estimate in an iterative fashion. The underlying idea is that, by introducing latent variables, the loglikelihood in Eq. S.2 can be decomposed into two terms that can be iteratively optimized, yielding the so-called E-step and M-step of the EM algorithm. We first derive the full EM updates for our model setup. We then show how these updates can be simplified in the incremental EM setup, a variant of EM that is more appropriate to our setting (see below for a discussion of our choice of EM algorithm).

1.2 Classical EM for similarity-based novelty

Latent variables. To apply the EM algorithm to our mixture model, we define vectors of binary latent random variables $\mathbf{z}_{t'} = (z_{1,t'}, ..., z_{N,t'})^{\mathrm{T}}$ for each time step $t' = 1, ..., t$, such that the following three conditions hold:

$$\Pr\left(z_{j,t'} = 1|\mathbf{w}\right) = w_j \qquad \text{for all } j = 1, ..., N \text{ and } t' = 1, ..., t \tag{S.5}$$

$$\Pr\left(s_{t'}|z_{j,t'} = 1\right) = k_j(s_{t'}) \qquad \text{for all } j = 1, ..., N \text{ and } t' = 1, ..., t \tag{S.6}$$

$$\sum_{j=1}^{N} z_{j,t'} = 1 \qquad \text{for all } t' = 1, ..., t. \tag{S.7}$$

Note that the last condition, Eq. S.7, implies that each vector $\mathbf{z}_{t'}$ is a one-hot-coding vector of the N component indices. The probability of a given $\mathbf{z}_{t'}$ to be a one-hot-coding vector of index j is given by the component weight w_j (Eq. S.5), while the probability of observing stimulus $s_{t'}$ when $\mathbf{z}_{t'}$ is encoding index j is the value of component j at stimulus $s_{t'}$ (Eq. S.6). Effectively, the latent variables thus link the component activations for a given stimulus and the component weights.

Eq. S.5 and Eq. S.6 allow us to write conditional distributions for the stimulus and latent variable at time t', $s_{t'}$ and $\mathbf{z}_{t'}$:

$$p(s_{t'}|\mathbf{z}_{t'}) = \prod_{j=1}^{N} (k_j(s_{t'}))^{z_{j,t'}} \tag{S.8}$$

$$p(\mathbf{z}_{t'}|w) = \prod_{j=1}^{N} (w_j)^{z_{j,t'}}. \tag{S.9}$$

We consider the observations $s_{t'}$ to be independent (see discussion below). That allows us to further write the joint distributions

$$p(s_{1:t}|\mathbf{z}_{1:t}) = \prod_{t'=1}^{t} P(s_{t'}|\mathbf{z}_{t'}) = \prod_{t'=1}^{t}\prod_{j=1}^{N} (k_j(s_{t'}))^{z_{j,t'}} \tag{S.10}$$

$$p(\mathbf{z}_{1:t}|w) = \prod_{t'=1}^{t} P(\mathbf{z}_{t'}|w) = \prod_{t'=1}^{t}\prod_{j=1}^{N} (w_j)^{z_{j,t'}}. \tag{S.11}$$

These distributions will give us the explicit expressions for the E-step and M-step that we will now derive via the loglikelihood decomposition.

Loglikelihood decomposition. The latent variables $\mathbf{z}_t$ allow for a decomposition of the loglikelihood in Eq. 7. Let $q(\mathbf{z}_{1:t})$ be a arbitrary distribution over the latent variables $\mathbf{z}_{1:t}$, then we can write the loglikelihood as

$$\log \Pr(s_{1:t}|\mathbf{w}) \Pr(\mathbf{w}) \overset{\text{law of total prob.}}{=} \sum_{\mathbf{z}_{1:t}} q(\mathbf{z}_{1:t}|\mathbf{w}) \log \Pr(s_{1:t}|\mathbf{w}) + \log \Pr(\mathbf{w}) \tag{S.12}$$

$$= \sum_{\mathbf{z}_{1:t}} q(\mathbf{z}_{1:t}|\mathbf{w}) \log \left(\frac{\Pr(s_{1:t}, \mathbf{z}_{1:t}|\mathbf{w})}{\Pr(\mathbf{z}_{1:t}|s_{1:t}, \mathbf{w})} \right) + \log \Pr(\mathbf{w}) \tag{S.13}$$

$$= \sum_{\mathbf{z}_{1:t}} q(\mathbf{z}_{1:t}|\mathbf{w}) \log \left(\frac{\Pr(s_{1:t}, \mathbf{z}_{1:t}|\mathbf{w})}{q(\mathbf{z}_{1:t}|\mathbf{w})} \right) + \log \Pr(\mathbf{w}) - \sum_{\mathbf{z}_{1:t}} q(\mathbf{z}_{1:t}|\mathbf{w}) \log \left(\frac{\Pr(\mathbf{z}_{1:t}|s_{1:t}, \mathbf{w})}{q(\mathbf{z}_{1:t}|\mathbf{w})} \right) \tag{S.14}$$

$$= \mathcal{L}(q(\mathbf{z}_{1:t}), p(s_{1:t}, \mathbf{z}_{1:t})) + \mathrm{KL}(q(\mathbf{z}_{1:t})||p(\mathbf{z}_{1:t}|s_{1:t})), \tag{S.15}$$

where $p(\mathbf{z}_{1:t}|s_{1:t}) = \Pr(\mathbf{z}_{1:t}|s_{1:t}, \mathbf{w})$. Based on this decomposition, the loglikelihood can be maximized by iteratively (i) minimizing the KL divergence (E-step) while keeping the weights $\mathbf{w}$ fixed at their prior values, and (ii) maximizing the Q term with respect to $\mathbf{w}$ while keeping the latent distribution q fixed (M-step). For the proof that this iterative procedure indeed maximizes the loglikelihood, see e.g. Ref. [46, 71].

E-step. The KL-divergence of q and p becomes minimal when the two distributions are equal, so

in the E-step, we set

$$q(\mathbf{z}_{1:t}|\mathbf{w}) = p(\mathbf{z}_{1:t}|s_{1:t}, \mathbf{w}) \tag{S.16}$$

$$= \frac{\Pr(s_{1:t}|\mathbf{z}_{1:t}, \mathbf{w}^{(t)}) \Pr(\mathbf{z}_{1:t}|\mathbf{w})}{\Pr(s_{1:t}|\mathbf{w})} \tag{S.17}$$

$$= \prod_{t'=1}^{t} \prod_{j=1}^{N} \left(\frac{w_j k_j(s_{t'})}{\sum_{j'=1}^{N} w_{j'} k_{j'}(s_{t'})} \right)^{z_{j,t'}} \tag{S.18}$$

$$= \prod_{t'=1}^{t} \prod_{j=1}^{N} (\gamma_{j,t'})^{z_{j,t'}} , \tag{S.19}$$

where we have used the mutual independence of any two observations $s_{t'}$, $s_{t''}$ and any two latent variables $\mathbf{z}_{t'}$ and $\mathbf{z}_{t''}$, respectively to obtain Eq. S.17, and substituted the distributions from Eq. 3, Eq. S.8 and Eq. S.9 to obtain Eq. S.18. To compute the distribution q in the E-step, it is thus sufficient to compute the terms

$$\gamma_{j,t'} = \frac{w_j k_j(s_{t'})}{\sum_{j'=1}^{N} w_{j'} k_{j'}(s_{t'})}, \tag{S.20}$$

also called 'responsibilities', for every $j = 1, ..., N$ and every observed stimulus $s_{t'}$, $t' = 1, ..., t$.

M-step. To maximize the term $\mathcal{L}(q(\mathbf{z}_{1:t}), p(s_{1:t}, \mathbf{z}_{1:t}))$ with respect to $\mathbf{w}$, we keep the latent distribution q fixed (denoted by fixed weights $\mathbf{w}_{\text{E-step}}$ that parametrize q). Note that

$$\mathcal{L}(q(\mathbf{z}_{1:t}), p(s_{1:t}, \mathbf{z}_{1:t})) = \sum_{\mathbf{z}_{1:t}} q(\mathbf{z}_{1:t}|\mathbf{w}_{\text{E-step}}) \log \left(\frac{p(s_{1:t}, \mathbf{z}_{1:t}|\mathbf{w})}{q(\mathbf{z}_{1:t}|\mathbf{w}_{\text{E-step}})} \right) + \log \Pr(\mathbf{w}) \tag{S.21}$$

$$= \sum_{\mathbf{z}_{1:t}} q(\mathbf{z}_{1:t}|\mathbf{w}_{\text{E-step}}) \log p(s_{1:t}, \mathbf{z}_{1:t}|\mathbf{w}) + \log \Pr(\mathbf{w}) - \sum_{\mathbf{z}_{1:t}} q(\mathbf{z}_{1:t}|\mathbf{w}_{\text{E-step}}) \log q(\mathbf{z}_{1:t}|\mathbf{w}_{\text{E-step}}). \tag{S.22}$$

Since we keep the latent distribution q fixed throughout the maximization, the last term in Eq. S.22 is constant in $\mathbf{w}$. It is thus sufficient to maximize

$$Q(\mathbf{w}, \mathbf{w}^{(t)}) = \sum_{\mathbf{z}_{1:t}} q(\mathbf{z}_{1:t}|\mathbf{w}) \log p(s_{1:t}, \mathbf{z}_{1:t}|\mathbf{w}) + \log \Pr(\mathbf{w}) \tag{S.23}$$

$$= \sum_{\mathbf{z}_{1:t}} q(\mathbf{z}_{1:t}|\mathbf{w}) \log \Pr(s_{1:t}|\mathbf{z}_{1:t}, \mathbf{w}) \Pr(\mathbf{z}_{1:t}|\mathbf{w}) + \log \Pr(\mathbf{w}) \tag{S.24}$$

$$= \sum_{\mathbf{z}_{1:t}} \prod_{t'=1}^{t} \prod_{j=1}^{N} (\gamma_{j,t'})^{z_{j,t'}} \log \left(\prod_{t'=1}^{t} \prod_{j=1}^{N} (w_j k_j(s_{t'}))^{z_{j,t'}} \right) + \log \Pr(\mathbf{w}) \tag{S.25}$$

$$= \sum_{\mathbf{z}_{1:t}} \prod_{t'=1}^{t} \prod_{j=1}^{N} (\gamma_{j,t'})^{z_{j,t'}} \sum_{t'=1}^{t} \sum_{j=1}^{N} z_{j,t'} \log (w_j k_j(s_{t'})) + \log \Pr(\mathbf{w}), \tag{S.26}$$

where we have substituted the distributions in Eq. S.8, Eq. S.9 and the expression for q from the E-step. Since all $\mathbf{z}_{t'}$ are one-hot coding vectors, exactly one $z_{t',j}$ is non-zero for each t'; and since we are summing over all possible one-hot coding sequences of vectors $\mathbf{z}_{1:t}$, each $z_{j,t'}$ will be non-zero exactly once. Eq. S.26 therefore reduces to

$$Q(\mathbf{w}, \mathbf{w}^{(t)}) = \sum_{t'=1}^{t} \sum_{j=1}^{N} \gamma_{j,t} (\log k_j(s_{t'}) + \log w_j) + \log \Pr(\mathbf{w}). \tag{S.27}$$

We maximize Q as in Eq. S.27 under the constraint that all weights sum to one using Lagrange multipliers, by solving

$$0 = \frac{\mathrm{d}}{\mathrm{d}w_j}\left(Q(\mathbf{w}, \mathbf{w}^{(t)}) - \lambda\left(\sum_{j'=1}^{N} w_{j'} - 1\right)\right) = \frac{1}{w_j}\sum_{t'=1}^{t}\gamma_{j,t'} + \frac{\mathrm{d}}{\mathrm{d}w_j}\log \Pr(\mathbf{w}) - \lambda \tag{S.28}$$

for all $j = 1, ..., N$. Since the Lagrange multiplier has to satisfy Eq. S.28 for all j, it has to satisfy the following equation (after multiplying Eq. S.28 with w_j and summing over j):

$$\begin{aligned}
0 &= \sum_{j=1}^{N}\sum_{t'=1}^{t}\gamma_{j,t'} + \sum_{j=1}^{N} w_j \frac{\mathrm{d}}{\mathrm{d}w_j}\log \Pr(\mathbf{w}) - \lambda\sum_{j=1}^{N} w_j && \text{(S.29)}\\
&= \sum_{t'=1}^{t}\sum_{j=1}^{N}\gamma_{j,t'} + \sum_{j=1}^{N} w_j \frac{\mathrm{d}}{\mathrm{d}w_j}\log \Pr(\mathbf{w}) - \lambda\sum_{j=1}^{N} w_j && \text{(S.30)}\\
&\overset{\text{Eq. S.7}}{=} t + \sum_{j=1}^{N} w_j \frac{\mathrm{d}}{\mathrm{d}w_j}\Pr(\mathbf{w}) - \lambda, && \text{(S.31)}
\end{aligned}$$

yielding

$$\lambda = t + \sum_{j=1}^{N} w_j \frac{\mathrm{d}}{\mathrm{d}w_j}\log \Pr(\mathbf{w}). \tag{S.32}$$

Substitution in Eq. S.28 gives the solution for the new mixture weights:

$$w_j = \frac{1}{t + c_j(w)}\sum_{t'=1}^{t}\gamma_{j,t'}, \tag{S.33}$$

where

$$c_j(w) = \sum_{j' \neq j} w_{j'} \frac{\mathrm{d}}{\mathrm{d}w_{j'}}\log \Pr(\mathbf{w}) \tag{S.34}$$

Special Dirichlet priors. For our Dirichlet prior (Eq. S.3), $c_j(w)$ reduces to

$$\begin{aligned}
c_j(w) &= \sum_{j' \neq j} w_{j'} \frac{\mathrm{d}}{\mathrm{d}w_{j'}}\log \Pr(\mathbf{w}) && \text{(S.35)}\\
&= \sum_{j' \neq j} w_{j'} \frac{\mathrm{d}}{\mathrm{d}w_{j'}}\left(\log c(\alpha) + \sum_{j''=1}^{N}(\alpha_{j''} - 1)\log w_{j''}\right) && \text{(S.36)}\\
&= \sum_{j' \neq j}(\alpha_{j'} - 1). && \text{(S.37)}\\
& && \text{(S.38)}
\end{aligned}$$

For Dirichlet prior with shared concentration parameter $\alpha_j = \epsilon + 1$ for all $j = 1, ..., N$, $c_j(w)$ reduces to $(N-1)\epsilon$ for all j. In this case, the M-step update (Eq. S.33) reduces to

$$w_j = \frac{1}{t + (N-1)\epsilon}\sum_{t'=1}^{t}\gamma_{j,t'}. \tag{S.39}$$

A Dirichlet prior with shared concentration parameter encodes the assumption that all components have been observed equally prior to observing the stimulus sequence $s_{1:t}$ – a reasonable assumption that we will make for the rest of the derivation.

1.3 Incremental EM for similarity-based novelty

Full EM. In summary, the full EM algorithm estimates the mixture weights that maximize the likelihood of our observations $s_{1:t}$ in two alternating steps: first, the 'responsibilities' $\gamma_{j,t'}$ are updated for all $j = 1, ..., N$ and $t' = 1, ..., t$ as in Eq. S.20 (E-step); then the weights $w_j^{(t)}$ after observations $s_{1:t}$ are computed as in Eq. S.39 (M-step). In the full EM setup, these two update steps are repeated until convergence of the algorithm. Since the full EM relies on knowledge about all observations $s_{1:t}$ to make a single update, the mixture weights can only be estimated after observing the full sequence $s_{1:t}$ that is then processed together in every iteration of the EM algorithm ('batch mode').

Incremental EM (I-EM). For our goal of modeling human and animals, however, this 'batch' update of the familiarity density (and hence, of stimulus novelty) is undesired since we assume that the brain updates its estimate of the novelty of a stimulus immediately after its observation – even if that leads to a less precise estimate of the overall statistics of stimuli in the environment. Instead of the full EM algorithm, we therefore estimate our component weights using the incremental EM algorithm. This variant of EM that uses only a single observation in each update iteration, iterating through the sequence of observations $s_{1:t}$ one-by-one, and then starting again at the beginning of the sequence until the algorithm has converged.

Continual incremental EM (CI-EM). For our scenario of humans and animals experiencing an unfamiliar environment, we assume that the incremental EM algorithms uses a continuing sequence of observations for its estimates, such that each observation is used only once for the update of the algorithm. The resulting weight estimates are in that sense 'approximate' estimates of the component weights but also the best estimates that are available for a potentially infinite sequence of observations. Moreover, in environments with finitely many distinguishable stimuli, all stimuli will eventually be revisited and thus contribute again to the weight estimate (like in the incremental EM for finite observation sequences). An important difference is, thought, that the proportion of how much each stimulus is revisited and thus contributes to the estimate of the familiarity distribution is not determined by a fixed circular schedule but by the experimenter (passive viewing task) or by the human or animal through its action policy (active exploration). To distinguish our specific use of the incremental EM algorithm from its classical 'finite stimulus sequence' usage, we also refer to our algorithm as 'continual incremental EM algorithm'.

Continual incremental EM (CI-EM) updates. The main difference between full EM and incremental EM updates is that the latter only uses a single observation $s_{t'}$ in its E-step update: instead of recomputing the responsibilities for all latent variables $z_{j,t''}$, for all $j = 1, ..., N$ and all $t'' = 1, ..., t$, the incremental E-step only recomputes the responsibilities for the observation $s_{t'}$; all responsibilities from previous iterations of the algorithm are kept the same:

$$\gamma_{j,t''}^{(i)} = \begin{cases} \frac{w_j^{(i-1)} k_j(s_{t''})}{\sum_{j'=1}^{N} w_{j'}^{(i-1)} k_{j'}(s_{t''})} & \text{if } t'' = t' \\ \gamma_{j,t''}^{(i-1)} & \text{else.} \end{cases} \tag{S.40}$$

where the additional index i denotes the ith iteration of the algorithm, in which observation $s_{t'}$ is considered. The M-step of the algorithm is the same as in the full EM algorithm (Eq. S.39) – just the underlying responsibilities are different due to the incremental E-step.

In our continual I-EM algorithm, each observation s_t is followed by one iteration of the I-EM

algorithm, i.e. the t-th step of algorithm considers the t-th observation:

$$\gamma_{j,t'}^{(t)} = \begin{cases} \frac{w_j^{(t-1)} k_j(s_{t'})}{\sum_{j'=1}^{N} w_{j'}^{(t-1)} k_{j'}(s_{t'})} & \text{if } t' = t \\ \gamma_{j,t'}^{(t-1)} & \text{else.} \end{cases} \tag{S.41}$$

In this scenario, the t-th M-step update (Eq. S.39) can be written as

$$w_j^{(t)} = \frac{1}{t + (N-1)\epsilon} \sum_{t'=1}^{t} \gamma_{j,t'}^{(t)} = \frac{1}{t + (N-1)\epsilon} \sum_{t'=1}^{t} \gamma_{j,t'}^{(t')}, \tag{S.42}$$

where $w_j^{(t')}$ for $j = 1, ..., N$ are the component weights at time t' in the experiment. Eq. S.42 gives rise to an iterative update rule (also see Methods, where we dropped the iteration index t' for better readability):

$$\begin{aligned} w_j^{(t)} &= \frac{1}{t + \epsilon(N-1)} \sum_{t'=1}^{t} \gamma_{j,t'}^{(t')} \\ &= \frac{1}{t + \epsilon(N-1)} \left(\gamma_{j,t}^{(t)} + \sum_{t'=1}^{t-1} \gamma_{j,t'}^{(t')} \right) \\ &= \frac{1}{t + \epsilon(N-1)} \left(\gamma_{j,t}^{(t)} + (t - 1 + \epsilon(N-1)) w_j^{(t-1)} \right) \\ &= w_j^{(t-1)} + \frac{1}{t + \epsilon(N-1)} \left(\gamma_{j,t}^{(t)} - w_j^{(t-1)} \right), \end{aligned} \tag{S.43}$$

where $\gamma_{j,t}^{(t)}$ is given by Eq. S.41.

1.4 Non-leaky count-based novelty as a special case of similarity-based novelty

In a scenario where all observations are (perfectly) distinct, count-based novelty and similarity-based novelty are equivalent. To formalize this statement, consider a sequence of observations $s_{1:t}$, where each observation is drawn out of a set of N perfectly distinct stimuli, $S = \{s_1, ..., S_N\}$. We say that count-based novelty and similarity-based novelty are equivalent with respect to this sequence if, at all time steps $t' = 0, ..., t$ and for all stimuli $s_j \in S$, their familiarity distributions $p_c^{(t')}$ (count-based) and $p_k^{(t')}$ (similarity-based) take identical values, i.e.

$$p_c^{(t')}(s_j) = p_k^{(t')}(s_j) \quad \text{for all } t' = 0, ..., t \text{ and all } j = 1, ..., N. \tag{S.44}$$

We now show that Eq. S.44 holds for count-based and similarity-based novelty. To this end, first note that the count-based familiarity at each time point t is given as

$$p_c^{(t)}(s) = \sum_{j=1}^{N} \delta(s, s_j) \frac{C_t(s_j) + \epsilon_c}{t + N\epsilon_c}, \tag{S.45}$$

where ϵ_c is the count-based prior and δ is the Kronecker delta function. We can rewrite the count-based familiarity distribution (Eq. S.45) in the style of a similarity-based familiarity density (i.e.

as a mixture model) as follows:

$$p_c^{(t)}(s) = \sum_{j=1}^{N} \delta(s, s_j)\tilde{w}_j^{(t)}, \tag{S.46}$$

with

$$\tilde{w}_j^{(t)} = \frac{C_t(s_j) + \epsilon_c}{t + N\epsilon_c}. \tag{S.47}$$

Since the counts are initialized as $C_0(s_j) = 0$ for all $j = 1, ..., N$, the initial 'count-based' mixture weights at time $t = 0$ are given as

$$\tilde{w}_j^{(0)} = \frac{1}{N}, \tag{S.48}$$

for all stimuli $j = 1, ..., N$. Since counts are updated at subsequent time steps t as

$$C_t(s_j) = C_{t-1}(s_j) + \delta(s_t, s_j), \tag{S.49}$$

the update of the count-based mixture weights is given as:

$$\begin{aligned}
\tilde{w}_j^{(t)} &= \frac{C_t(s_j) + \epsilon_c}{t + N\epsilon_c} && \text{(S.50)}\\
&\overset{\text{Eq. S.49}}{=} \frac{C_{t-1}(s_j) + \epsilon_c}{t + N\epsilon_c} + \frac{\delta(s_t, s_j)}{t + N\epsilon_c} && \text{(S.51)}\\
&= \frac{C_{t-1}(s_j) + \epsilon_c}{t - 1 + N\epsilon_c}\frac{t - 1 + N\epsilon}{t + N\epsilon} + \frac{\delta(s_t, s_j)}{t + N\epsilon_c} && \text{(S.52)}\\
&= \tilde{w}_j^{(t-1)}\left(1 - \frac{1}{t + N\epsilon}\right) + \frac{\delta(s_t, s_j)}{t + N\epsilon_c}. && \text{(S.53)}
\end{aligned}$$

We now construct a similarity-based novelty model that is equivalent to the previous count-based novelty model. Since all observations s_j are perfectly distinct, we can construct a continuous stimulus space $S' \supset S$ and non-overlapping, box-shaped components k_j, $j = 1, ..., N$, such that

$$k_j(s) = \begin{cases} 1 & \text{if } s = s_j, \\ 0 & \text{otherwise.} \end{cases} \tag{S.54}$$

and $\int_{S'} k_j(s)\,\mathrm{d}s$ for all $j = 1, ..., N$ (e.g., we can place the s_j on a real line with equal distance 1 between them, and define box components of height and width 1, centered at the different s_j). Then the familiarity distribution of the similarity-based novelty model at stimulus $s_i \in S$ is given as

$$p_k^{(t)}(s_i) = \sum_{j=1}^{N} \delta(s_i, s_j) w_j^{(t)}, \tag{S.55}$$

By comparing with Eq. S.46, we can see that, for the equality in Eq S.44 to hold, it is sufficient to show that $\tilde{w}_j^{(t)} = w_j t$ for all j and all t. The component weights are uniformly initialized to $w_j^{(0)} = \frac{1}{N}$, and thus equal to the count-based initial weights (Eq. S.48). The component weights are updated with the incremental update rule derived above (Eq. S.43), i.e.

$$\begin{aligned}
w_j^{(t)} &= w_j^{(t-1)} + \frac{1}{t + \epsilon_k(N-1)}\left(\gamma_{j,t}^{(t)} - w_j^{(t-1)}\right) && \text{(S.56)}\\
&= w_j^{(t-1)}\left(1 - \frac{1}{t + \epsilon_k(N-1)}\right) + \frac{1}{t + \epsilon_k(N-1)}\gamma_{j,t}^{(t)} && \text{(S.57)}\\
&= w_j^{(t-1)}\left(1 - \frac{1}{t + \epsilon_k(N-1)}\right) + \frac{\delta(s_t, s_j)}{t + \epsilon_k(N-1)}, && \text{(S.58)}
\end{aligned}$$

where we have used that, given our choice of box components, we have

$$\gamma_{j,t}^{(t)} = \frac{w_j^{(t)} \delta(s_t, s_j)}{\sum_{j'=1} w_{j'}^{(t)} \delta(s_t, s_{j'})} = \delta(s_t, s_j). \tag{S.59}$$

By comparing with the update rule of the count-based mixture weights (Eq. S.53), we can see that by setting the similarity-based novelty prior $\epsilon_k = \frac{N}{N-1}$, the two weight updates become equivalent, and, hence, the two novelty definitions are equal in the sense of Eq. S.44.

1.5 Network implementation of similarity-based novelty

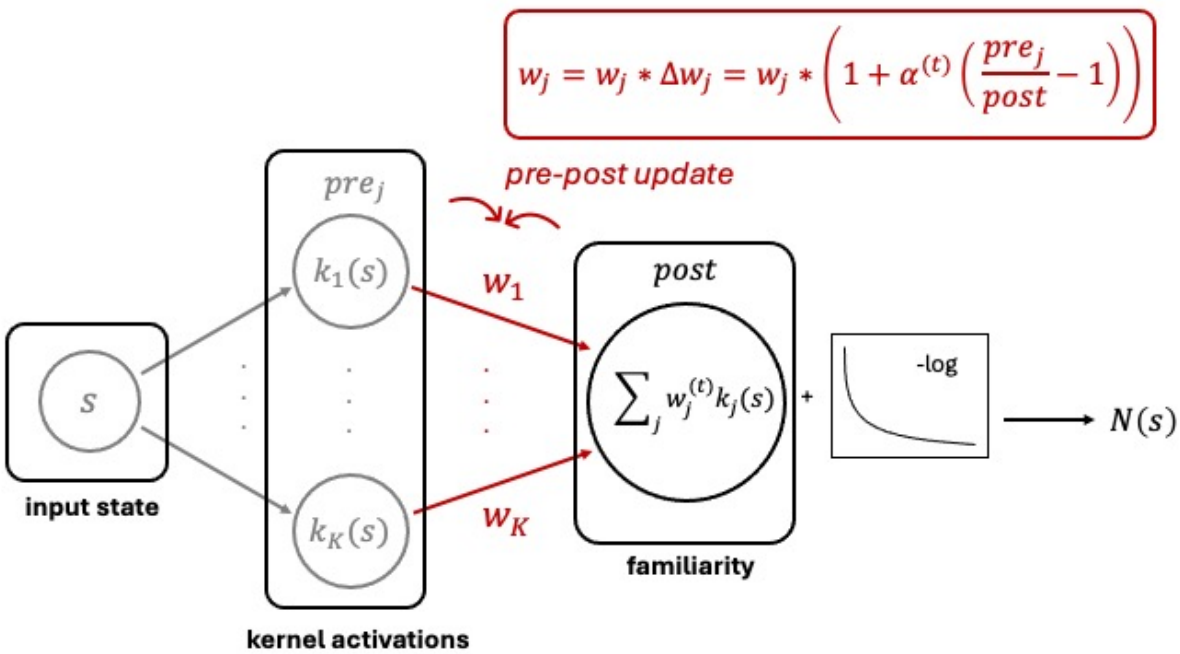

Figure 5: Network implementation of similarity-based novelty. The weight update rule of similarity-based novelty (Eqs. 4,5 in main text) can be written as a multiplicative weight update rule (red box) in a biologically realistic circuit. The amplitude of weight updates depends on the ratio of presynaptic activity (component activation by current input stimulus s) and postsynaptic activity (familiarity of current input stimulus s, given the previous weights).

2 Passive viewing experiment: data set and modeling

2.1 Extraction of neural responses in Homann et al.

Population responses are extracted from the raw fluorescence traces in six steps: (i) fluorescence traces $F(t)$ for each ROI (manually identified) are obtained by averaging over the ROI's pixels on a frame-by-frame basis; (ii) relative fluorescence traces for each ROI are computed as $\Delta F/F(t) = \frac{F(t)-F_0(t)}{F_0(t)}$, where a smooth estimate of the baseline fluorescence $F_0(t)$ is computed using spline interpolation; (iii) event-triggered activity (ETA) traces for each event type (novel image, familiar image, recovery probe) is computed by trial-averaging relative fluorescence traces $\Delta F/F(t)$ in the window 1000 ms before until 300-500 ms after the event, separately for each ROI; (iv) excess activity traces in each ROI for the novel image and the recovery probe events are obtained by subtracting the ETA for familiar image events from the respective ETAs of novel image and

recovery probe events; (v) population excess activity traces (and population steady-state activity traces) are computed as the population average over the individual ROI's excess (and steady-state) activity traces; (vi) the population's novelty and recovery probe responses ΔN are given by the maximum amplitude of the population excess activity to the respective event (novel image or recovery probe), while the population's steady-state response N_∞ to familiar images is calculated as the average of the population steady-state activity trace.

2.2 Grid search: parameter stability

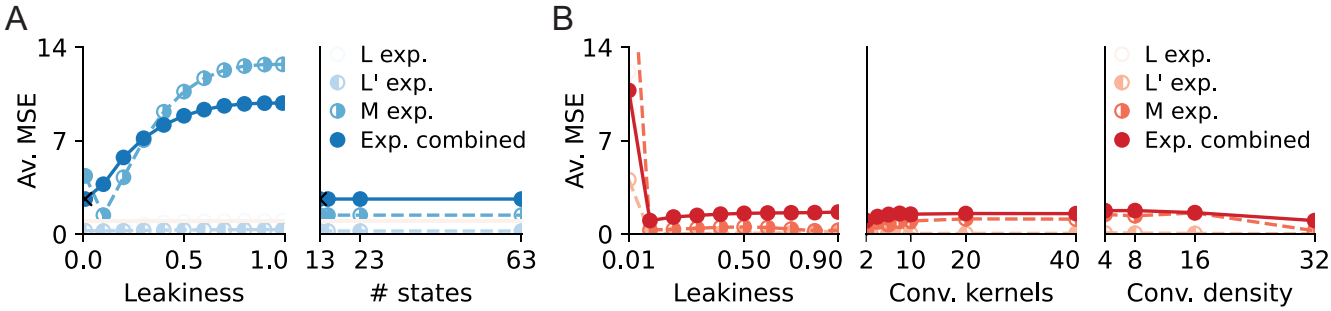

Figure 6: Robustness of fit to neural data. Individual MSE for each experiment by Homann et al. [38] and average MSE across experiments for different values of (A) the count-based novelty parameters α (leakiness) and $|S|$ (number of states; and (B) the similarity-based novelty parameters α (leakiness), c (number of different features encoded by components) and d (number of convolutional components for each feature).

2.3 Linear regression fit to neural data

For each parameter set θ of a given model $\mathcal{M}$, we use least-squares estimation (LSE) to find the regression coefficients β that minimize the regression error ϵ in the regression equation

$$Y = \beta X + \epsilon, \tag{S.60}$$

where Y is the neural data vector, and X contains the model predictions for Y. The data vector $Y \in \mathbb{R}^{20}$ contains the average neural responses to novelty (and familiarity) measured in the three experiment variations:

$$Y = \left(y_1^{(1)}, ..., y_5^{(1)}, y_1^{(2)}, ..., y_7^{(2)}, y_1^{(3)}, ..., y_4^{(3)}, y_1^{(4)}, ..., y_4^{(4)}\right) \tag{S.61}$$

where $y_i^{(1)}$ are the average neural novelty responses ΔN during the 'variable repetition experiment' for different number of repetitions L of the familiar sequence ($L = 1, 3, 8, 18, 38$); $y_i^{(2)}$ are the average neural novelty responses ΔN to the formerly familiar sequence in the 'repeated image set experiment', for different lengths ΔT of the replacement interval ($\Delta T = 0, 21, 42, 63, 84, 108, 144$ seconds); and $y_i^{(3)}$ and $y_i^{(4)}$ are the average neural responses to novelty, ΔN, and to familiarity, N_∞, respectively, during the 'variable image number experiment' for different number M of familiar stimuli ($M = 3, 6, 9, 12$). The model predictions $X \in \mathbb{R}^{20\times 2}$ are given as

$$X = \begin{pmatrix} 0 & \dots & 0 & 1 & \dots & 1 \\ x_1^{(1)} & \dots & x_4^{(3)} & x_1^{(4)} & \dots & x_4^{(4)} \end{pmatrix}, \tag{S.62}$$

where the row column, containing the model's respective predictions for the novelty and familiarity signals in each of the experimental conditions underlying the data vector Y, is used to fit the scaling factor β_2 (second entry of $\beta \in \mathbb{R}^2$) of the linear regression, and the first row is used to fit the shift β_1 (first entry of $\beta \in \mathbb{R}^2$) of the predicted steady-state responses N_∞ relative to the actual neural signals to familiar stimuli. Note that since the novelty responses ΔN are normalized with

the steady-state responses, the shift does not apply to them, such that they have zero-entries in the first row of X.

For each model $\mathcal{M}$, we choose the parameter set θ that minimizes the mean squared error (MSE) between neural data Y and the fitted model predictions βX^*, normalized with the variance of the neural data $\sigma^2(Y)$:

$$\epsilon_{\text{regr}} = \frac{\text{MSE}(Y, \beta X^*)}{\sigma^2(Y)}. \tag{S.63}$$

3 Active exploration experiment: data set and modeling

3.1 Behavioral data by Rosenberg et al.: Outliers

For mouse 'B2', which fails to receive a reward during its first 6 visits to the goal state, we only consider behavior up to its first goal encounter. Like Rosenberg et al., we exclude mouse 'D6' since it fails to enter the maze or more than one short (1 time step) bout. In contrast to Rosenberg et al., we include mouse 'C6' in our analysis since its behavior with respect to our behavioral statistic does not show outlier behavior. While Rosenberg et al. analyze exploration behavior based on 'first water runs', which includes the direct paths between maze entry and the goal state in the maze, we analyze the entire state trajectory of mice during the exploration between first entry and encounter of the goal state.

3.2 Novelty-seeking RL (N-RL) algorithms

3.2.1 Model-based N-RL algorithm

The model-based (MB) N-RL algorithm (Alg. 1) uses prioritized sweeping (e.g., see [74]) for the model-based update of the novelty-based Q-values ('N-Q-values'), and applies a softmax policy on the N-Q-values to determine its actions.

The MB N-RL algorithm is characterized by the parameters of the novelty model (ϵ_c for count-based novelty; ϵ_k and $\mathbf{k}$ for similarity-based novelty, see Normative update rule for similarity-based novelty), as well as five additional parameters (see line 2 in Alg. 1). The inverse temperature β determines the noise level of the agent's softmax policy over the Q-values (low β leads to more noisy actions). The leak factor k_{leak} and the prior ϵ_{env} of the belief counts α determine how strongly the α values for a given transition decay between two observations of that transition - the higher k_{leak}, the more influenced are the α values by recently observed transitions, and the lower the influence of the prior ϵ_{env}. A lower ϵ_{env}, in turn, causes the α values to adjust more quickly to the observed transition counts. The more deterministic our environment is, the lower our ϵ_{env} should thus be, since every observation contains reliable information about the underlying state transition. The parameter λ and T_{PS} characterize the Q-value updates: T_{PS} determines how many sweeps, i.e. Q-value updates are performed in each step (the higher T_{PS}, the more precise the model-based Q-values update), while λ has a similar role to the discount factor γ that discounts the influence of future expected novelty on the state value of a given state s (the higher λ, the more important the value of future novelty).

3.2.2 Model-free N-RL algorithm

The model-free (MF) N-RL algorithm (Alg. 3) implements an actor-critic algorithm [74] that updates novelty-based V-values using TD-learning and novelty-based action preferences using policy gradient learning. Actions are chosen using a softmax policy over the action preferences.

In addition to the parameters of the novelty model (same as above for the MB N-RL algorithm), the MF N-RL algorithm is characterized by 8 parameters: the inverse temperature β of the softmax action policy, the initial V-values V_0 and action preferences h_0, the learning rates for critic (α_c)

Algorithm 1 Model-based N-RL algorithm

Parameter specification
Specify environment parameters S, A, p, and $s_{\mathrm{G}}, T_{\max}$.
Specify algorithm parameters $\{\lambda, T_{\mathrm{PS}}, k_{\mathrm{leak}}, \epsilon_{\mathrm{env}}, \beta\}$.
Specify novelty parameters $\{\epsilon_c\}$ (count-based novelty) or $\{\epsilon_k, \mathbf{k}\}$ (similarity-based novelty).
Variable initialization
Initialize novelty variables using Alg. 5 (count-based novelty) or Alg. 8 (similarity-based novelty).
Initialize $U^{(1)}(s) = \frac{N^{(1)}(s)}{1-\lambda}$ for all $s \in S$.
Initialize Q-values as $Q^{(1)}_{MB}(s,a) = U^{(1)}(s)$ for all $(s,a) \in S \times A$.
Initialize $\alpha^{(1)}_{s,a}(s') = \begin{cases} \epsilon_{\mathrm{env}} & \text{if } p(s,a,s') > 0 \\ \mathrm{nan} & \text{if } p(s,a,s') = 0 \end{cases}$ for all $(s,s',a) \in S \times S \times A$.
Initialize agent.
Set $t \leftarrow 0$ and initialize state $S_1 = s_1$.
Update novelty variables using Alg. 7 (count-based novelty) or Alg. 10 (similarity-based novelty).
Initialize time $t \leftarrow 1$.
# Simulate agent.
while $S_t \neq s_{\mathrm{G}}$ and $t < T_{\max}$ **do**
# Take action
  Sample action a_t from $\pi(A_t = a | S_t = s) \propto \exp\Big(\beta Q^{(t)}_{MB}(s,a)\Big)$.
  Evaluate action a_t in environment and observe $S_{t+1} = s_{t+1}$.
Update internal variables
  Update novelty variables using Alg. 7 (count-based novelty) or Alg. 10 (similarity-based novelty).
  Update $\alpha^{(t+1)}_{s,a}(s') = k_{\mathrm{leak}} \alpha^{(t)}_{s,a}(s') + (1 - k_{\mathrm{leak}})\epsilon_{\mathrm{env}}$ for all $(s,a,s') \in S \times A \times S$.
  Update $\alpha^{(t+1)}_{s_t,a_t}(s') = \alpha^{(t)}_{s_t,a_t}(s') + \delta(s', s_{t+1})$ for all $s' \in S$.
  Update beliefs $\hat{\theta}^{(t+1)}(s') = \alpha^{(t+1)}_{s,a}(s') / \sum_{\tilde{s}' \in S} \alpha^{(t+1)}_{s,a}(\tilde{s}')$.
  Compute $N^{(t+1)}(s)$ using Alg.6 (count-based novelty) or Alg. 9 (similarity-based novelty)
  Compute new Q-values $Q^{(t+1)}_{MB}(s,a)$ and new U-values $U^{(t+1)}(s)$ for all $(s,a) \in S \times A$, using $N^{(t+1)}(s)$ and Alg. 2
Update time step
  $t \leftarrow t + 1$.
end while

and actor (α_a), the discount factor for future novelty γ, and the two decay factors of the critic and actor eligibility traces λ and $\tilde{\lambda}$.

3.2.3 Hybrid N-RL algorithm

The hybrid N-RL algorithm runs the model-based and the model-free N-RL updates in parallel and uses a hybrid policy that is based on a linear combination of the model-based novelty Q-values and the model-free novelty action preferences.
The hybrid N-RL model has one set of novelty parameters, analogously to the MF and MB N-RL algorithms above, as well as the respective MF and MB algorithm parameters from Algs. 1,3. The only additional parameter is the weight w_{MF} that determines the balance of the MF and MB softmax distribution in the hybrid policy.

Algorithm 2 Prioritized Sweeping with target signal $N^{(t+1)}$

Initialize next Q-values $Q^{(t+1)}(s,a) = \sum_{s' \in S} \hat{\theta}^{(t+1)}_{s,a} s'(N^{(t+1)}(s') + \lambda U^{(t)}(s'))$ for all $(s,a) \in S \times A$.
Initialize next U-values $U^{(t+1)}(s) = U^{(t)}(s)$ for all $s \in S$.
Initialize priority queue $q = \left\{ \left| \max_{a \in A} Q^{(t+1)}(s,a) - U^{(t+1)}(s) \right| \right\}_{s \in S}$.
# Perform prioritized Q-value update
for $\tau = 1, ..., T_{PS}$ **do**
  Determine state with largest U-value update as $s_\tau = \arg\max(q)$.
  Compute U-value update for s_τ as $\Delta U^{(t+1)}_\tau = \max_{a \in A} Q^{(t+1)}(s_\tau, a) - U^{(t+1)}(s_\tau)$.
  Update $U^{(t+1)}(s_\tau) \leftarrow U^{(t+1)}(s_\tau) + \Delta U^{(t+1)}_\tau = \max_{a \in A} Q(s_\tau, a)$.
  Update next Q-values $Q^{(t+1)}(s,a) \leftarrow Q^{(t+1)}(s,a) + \lambda \hat{\theta}^{(t+1)}(s,a,s_\tau) \Delta U^{(t+1)}_\tau$ for all $(s,a) \in S \times A$.
  Recompute priority queue $q \leftarrow \left\{ \left| \max_{a \in A} Q^{(t+1)}(s,a) - U^{(t+1)}(s) \right| \right\}_{s \in S}$.
end for

Algorithm 3 Model-free N-RL algorithm

Parameter specification
Specify environment parameters S, A, p, s_G, $T_{\max}$.
Specify algorithm parameters $\{\lambda, \tilde{\lambda}, \gamma, \alpha_c, \alpha_a, \beta, V_0, h_0\}$.
Specify novelty parameters $\{\epsilon_c\}$ (count-based novelty) or $\{\epsilon_k, \mathbf{k}\}$ (similarity-based novelty).
Variable initialization
Initialize novelty variables using Alg. 5 (count-based novelty) or Alg. 8 (similarity-based novelty).
Initialize V-values as $V^{(1)}_{MF}(s) = V_0$ for all $s \in S$.
Initialize action preference values as $h^{(1)}_{MF}(s,a) = h_0$ for all $(s,a) \in S \times A$.
Initialize critic eligibility traces as $e^{(1)}(s) = 0$ for all $s \in S$.
Initialize actor eligibility traces as $\tilde{e}^{(1)}(s,a) = 0$ for all $(s,a) \in S \times A$.
Initialize agent.
Set $t \leftarrow 0$ and initialize state $S_1 = s_1$.
Update novelty variables using Alg. 7 (count-based novelty) or Alg. 10 (similarity-based novelty).
Initialize time $t \leftarrow 1$.
# Simulate agent.
while $S_t \neq s_{\mathrm{G}}$ and $t < T_{\max}$ **do**
# Take action
  Sample action a_t from $\pi(A_t = a | S_t = s) \propto \exp\left(\beta h^{(t)}_{MF}(s,a)\right)$.
  Evaluate action a_t in environment and observe $S_{t+1} = s_{t+1}$.
Update internal variables
  Update novelty variables using Alg. 7 (count-based novelty) or Alg. 10 (similarity-based novelty).
  Compute TD-error $\mathrm{TD}^{(t+1)} = N^{(t+1)}(s_{t+1}) - \gamma V^{(t)}_{MF}(s_{t+1}) - V^{(t)}_{MF}(s_t)$.
  Update critic eligibility traces $e^{(t+1)}(s) = \lambda e^{(t)}(s) + \delta(s, s_t)$ for all $s \in S$.
  Update actor eligibility traces $\tilde{e}^{(t+1)}(s,a) = \tilde{\lambda} e^{(t)}(s,a) + \delta(s, s_t)\delta(a, a_t)$ for all $(s,a) \in S \times A$.
  Update V-values $V^{(t+1)}_{MF}(s) = V^{(t)}_{MF}(s) + \alpha_c e^{(t+1)}(s) \mathrm{TD}^{(t+1)}$ for all $s \in S$.
  Update action preferences $h^{(t+1)}_{MF}(s,a) = h^{(t)}_{MF}(s,a) + \alpha_a \tilde{e}^{(t+1)}(s,a) \mathrm{TD}^{(t+1)}$ for all $(s,a) \in S \times A$.
Update time step
  $t \leftarrow t + 1$.
end while

Algorithm 4 Hybrid N-RL algorithm

Parameter specification

Specify environment parameters S, A, p, s_G, $T_{\max}$.

Specify algorithm parameters $\{\beta_{MB}, \lambda_{MB}, T_{\text{PS}}, k_{\text{leak}}, \epsilon_{\text{env}}, \beta_{MF}, \lambda_{MF}, \tilde{\lambda}_{MF}, \gamma, \alpha_c, \alpha_a, V_0, h_0\}$.

Specify novelty parameters $\{\epsilon_c\}$ (count-based novelty) or $\{\epsilon_k, \mathbf{k}\}$ (similarity-based novelty).

Variable initialization

Initialize novelty variables using Alg. 5 (count-based novelty) or Alg. 8 (similarity-based novelty).

Model-based variables

Initialize $U^{(1)}(s) = \frac{N^{(1)}(s)}{1-\lambda_{MB}}$ for all $s \in S$.

Initialize model-based Q-values as $Q^{(1)}_{MB}(s,a) = U^{(1)}(s)$ for all $(s,a) \in S \times A$.

Initialize $\alpha^{(1)}_{s,a}(s') = \begin{cases} \epsilon_{\text{env}} & \text{if } p(s,a,s') > 0 \\ \text{nan} & \text{if } p(s,a,s') = 0 \end{cases}$ for all $(s,s',a) \in S \times S \times A$.

Model-free variables

Initialize model-free V-values as $V^{(1)}_{MF}(s) = V_0$ for all $s \in S$.

Initialize model-free action preference values as $h^{(1)}_{MF}(s,a) = h_0$ for all $(s,a) \in S \times A$.

Initialize critic eligibility traces as $e^{(1)}(s) = 0$ for all $s \in S$.

Initialize actor eligibility traces as $\tilde{e}^{(1)}(s,a) = 0$ for all $(s,a) \in S \times A$.

Initialize agent.

Set $t \leftarrow 0$ and initialize state $S_1 = s_1$.

Update novelty variables using Alg. 7 (count-based novelty) or Alg. 10 (similarity-based novelty).

Initialize time $t \leftarrow 1$.

Simulate agent.

while $S_t \neq s_{\text{G}}$ and $t < T_{\max}$ **do**

Take action using hybrid policy

Sample action a_t from hybrid policy

$$\pi(A_t = a | S_t = s) \propto w_{MF} \exp\Big(\beta_{MF} h^{(t)}_{MF}(s,a)\Big) + (1 - w_{MF}) \exp\Big(\beta_{MB} Q^{(t)}_{MB}(s,a)\Big).$$

Evaluate action a_t in environment and observe $S_{t+1} = s_{t+1}$.

Update internal variables

Update novelty variables using Alg. 7 (count-based novelty) or Alg. 10 (similarity-based novelty).

Model-based update

Update $\alpha^{(t+1)}_{s,a}(s') = k_{\text{leak}} \alpha^{(t)}_{s,a}(s') + (1 - k_{\text{leak}}) \epsilon_{\text{env}}$ for all $(s,a,s') \in S \times A \times S$.

Update $\alpha^{(t+1)}_{s_t,a_t}(s') = \alpha^{(t)}_{s_t,a_t}(s') + \delta(s', s_{t+1})$ for all $s' \in S$.

Update beliefs $\hat{\theta}^{(t+1)}(s') = \alpha^{(t+1)}_{s,a}(s') / \sum_{\tilde{s}' \in S} \alpha^{(t+1)}_{s,a}(\tilde{s}')$.

Compute $N^{(t+1)}(s)$ using Alg.6 (count-based novelty) or Alg. 9 (similarity-based novelty)

Compute new Q-values $Q^{(t+1)}_{MB}(s,a)$ and new U-values $U^{(t+1)}(s)$ for all $(s,a) \in S \times A$, using $N^{(t+1)}(s)$ and Alg. 2

Model-free update

Compute TD-error $\text{TD}^{(t+1)} = N^{(t+1)}(s_{t+1}) - \gamma V^{(t)}_{MF}(s_{t+1}) - V^{(t)}_{MF}(s_t)$.

Update critic eligibility traces $e^{(t+1)}(s) = \lambda_{MF} e^{(t)}(s) + \delta(s, s_t)$ for all $s \in S$.

Update actor eligibility traces $\tilde{e}^{(t+1)}(s,a) = \tilde{\lambda}_{MF} e^{(t)}(s,a) + \delta(s,s_t)\delta(a,a_t)$ for all $(s,a) \in S \times A$.

Update V-values $V^{(t+1)}_{MF}(s) = V^{(t)}_{MF}(s) + \alpha_c e^{(t+1)}(s) \text{TD}^{(t+1)}$ for all $s \in S$.

Update action preferences $h^{(t+1)}_{MF}(s,a) = h^{(t)}_{MF}(s,a) + \alpha_a \tilde{e}^{(t+1)}(s,a) \text{TD}^{(t+1)}$ for all $(s,a) \in S \times A$.

Update time step

$t \leftarrow t + 1$.

end while

3.3 Novelty models for N-RL algorithm

Algorithm 5 Novelty initialization (count-based novelty)

Set $C^{(0)}(s) = 0$ for all $s \in S$.

Algorithm 6 Novelty evaluation (count-based novelty)

Compute novelty $N^{(t+1)}(s) = -\log \frac{C^{(t+1)}(s)+\epsilon}{t+|S|\epsilon}$ for all $s \in S$.

Algorithm 7 Novelty update (count-based novelty)

Update counts $C^{(t+1)}(s) = C^{(t)}(s) + \delta(s, S_{t+1})$ for all $s \in S$.

Algorithm 8 Novelty initialization (similarity-based novelty)

Set component weights $w_j^{(0)} = \frac{1}{K}$ for all $j = 1, ..., K$.

Algorithm 9 Novelty evaluation (similarity-based novelty)

Compute novelty $N^{(t+1)}(s) = -\log \sum_{j=1}^{K} w_j^{(t+1)} k_j(s)$ for all $s \in S$.

Algorithm 10 Novelty update (similarity-based novelty)

Compute responsibilities $\gamma_j^{(t+1)} = \frac{w_j^{(t)} k_j(S_{t+1})}{\sum_{j'=1}^{K} w_{j'}^{(t)} k_{j'}(S_{t+1})}$ for all $j = 1, ..., K$.

Update novelty weights $w_j^{(t+1)} = w_j^{(t)} + \frac{1}{t+K\tilde{\epsilon}} \left(\gamma_j^{(t+1)} - w_j^{(t)}\right)$.

3.4 Behavioral fit: MLE derivation for N-RL models

To compute the log-likelihood in Eq. 24 for a given N-RL model, we rewrite it in terms of the model's softmax policy $\pi(\cdot|\mathcal{M}, \theta)$ (see SI) as

$$\log P(\mathcal{D}|\mathcal{M}, \theta) = \sum_{i=1}^{20} \sum_{t=1}^{T_i} \log \pi(a_{t-1}^{(i)} | s_{t-1}^{(i)}; \mathcal{M}, \theta). \tag{S.64}$$

To compute the right-hand side of Eq. S.64, we simulate the model $\mathcal{M}$ with parameters θ for each mouse trajectory; but, instead of letting the model choose the action in each step t according to its policy, we choose the corresponding action $a_t^{(i)}$ from the mouse data and compute its log-likelihood under the model's policy. These individual log-likelihoods are then added to obtain the total likelihood of the data under the given model $\mathcal{M}$ with parameters θ (Eq. S.64).

The expression for the log-likelihood in Eq. S.64 can be derived as follows:

$$\log P(\mathcal{D}|\mathcal{M},\theta) \tag{S.65}$$

$$= \log \prod_{i=1}^{20} P(s_0^{(i)}, a_0^{(i)} ..., a_{T_1-1}^{(i)}, s_{T_1}^{(i)}|\mathcal{M},\theta) \tag{S.66}$$

$$= \log \prod_{i=1}^{20} P(s_0^{(i)}|\mathcal{M},\theta) P(a_0^{(i)}|s_0^{(i)};\mathcal{M},\theta) P(s_1^{(i)}|a_0^{(i)};\mathcal{M},\theta) ... P(s_{T_1}^{(i)}|a_{T_1-1}^{(i)};\mathcal{M},\theta) \tag{S.67}$$

$$= \sum_{i=1}^{20} \left(\log P(s_0^{(i)}|\mathcal{M},\theta) + \sum_{t=1}^{T_i} \log P(a_{t-1}^{(i)}|s_{t-1}^{(i)};\mathcal{M},\theta) + \sum_{t=1}^{T_i} \log P(s_t^{(i)}|a_{t-1}^{(i)};\mathcal{M},\theta) \right), \tag{S.68}$$

where we have use the Markov property of the N-RL models (see above). Note that $P(s_0^{(i)}|\mathcal{M},\theta) = 1$ since all mice and agents start in the home cage by experiment design; and that $P(s_{t+1}^{(i)}|a_t^{(i)};\mathcal{M},\theta) = 1$ since the environment is deterministic, i.e. given action a always leads to the same state s. Using this, we can simplify Eq. S.68 to

$$\log P(\mathcal{D}|\mathcal{M},\theta) = \sum_{i=1}^{20} \sum_{t=1}^{T_i} \log P(a_{t-1}^{(i)}|s_{t-1}^{(i)};\mathcal{M},\theta). \tag{S.69}$$

We can further write the individual log-likelihoods in Eq. S.69 in terms of the model's action policy $\pi(\cdot|\mathcal{M},\theta)$ (in the case of our N-RL models, a softmax policy over the model's Q-values):

$$\log P(a_{t-1}^{(i)}|s_{t-1}^{(i)};\mathcal{M},\theta) = \log \pi(a_{t-1}^{(i)}|s_{t-1}^{(i)}|\mathcal{M},\theta) \tag{S.70}$$

$$= \log \mathrm{softmax}(Q_{\mathcal{M},\theta}^{(t-1)}(s_{t-1}^{(i)}, a_{t-1}^{(i)})) \tag{S.71}$$

$$= \log \frac{\exp\left(\beta Q_{\mathcal{M},\theta}^{(t-1)}(s_{t-1}^{(i)}, a_{t-1}^{(i)})\right)}{\sum_{a'} \exp\left(\beta Q_{\mathcal{M},\theta}^{(t-1)}(s_{t-1}^{(i)}, a')\right)} \tag{S.72}$$

$$= \beta Q_{\mathcal{M},\theta}^{(t-1)}(s_{t-1}^{(i)}, a_{t-1}^{(i)}) - \log \sum_{a'} \exp\left(\beta Q_{\mathcal{M},\theta}^{(t-1)}(s_{t-1}^{(i)}, a')\right), \tag{S.73}$$

where the inverse temperature β of the softmax policy is part of the set of model parameters θ.

3.5 Behavioral fit: Model recovery

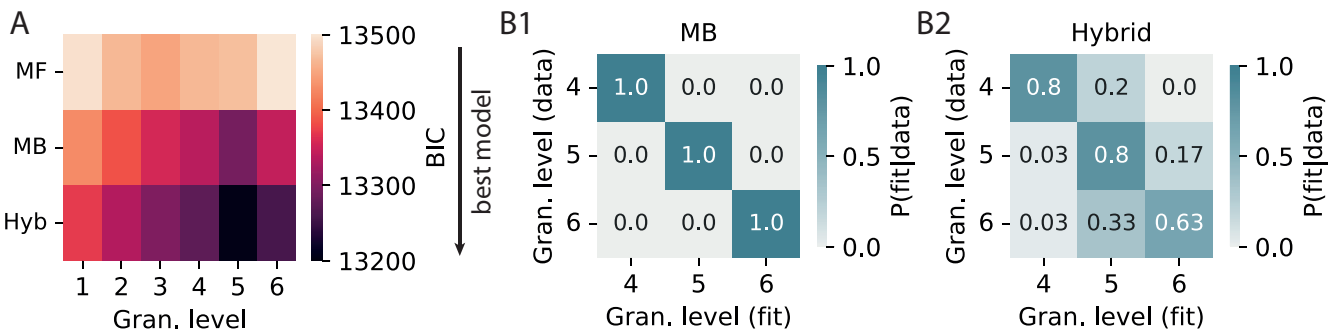

Figure 7: Model recovery for all similarity-based novelty models in the Rosenberg task.